NASA Contractor Report 185242

[redacted]

"Copy No. 27 "

P-177

High Speed Turboprop Aeroacoustic Study (Counterrotation)

Volume II – Computer Programs

C. E. Whitfield, R. Mani, and P.R. Gliebe
GE Aircraft Engines
Advanced Technology Operations
Cincinnati, Ohio 45215

July 1990

... in part. Date

for general release: August 1992.

Prepared for
Lewis Research Center
Under Contract NAS3-23721

National Aeronautics and Space Administration

(NASA-CR-185242) HIGH SPEED TURBOPROP AEROACOUSTIC STUDY (COUNTERROTATION). VOLUME 2: COMPUTER PROGRAMS (GE) 177 p

N92-31248

Unclas

G3/71 0116858

High Speed Turboprop Aeroacoustic Study (Counterrotation)

Volume II - Computer Programs

Table of Contents

Computer Programs

One major acoustic computer program and one performance preprocessor were developed under the counterrotation portion of this Contract described in Volume I.

CRPFAN (CounterRotation PropFan Noise) predicts

a. Steady loading and thickness noise of Rotor 1.

b. Steady loading and thickness noise of Rotor 2.

c. Unsteady loading noise of Rotor 2 resulting from the interaction of the blades with wakes and vortices shed from Rotor 1.

d. Additional BPF harmonic noise resulting from the interaction of the rotors with non–uniform inflow generated by the installation environment (see Volume I, Section 2.2). The installation effects considered are: angle–of–attack operation; the presence of a pylon; the presence of a fuselage, and the distortion behind a wing lifting line. This prediction is added to the results from (a) and (b) above.

The operation of the code is sketched in Figure 1. Depending on requirements, the program can be used to predict

Items (a) – (b);
(a) – (c);
(a) – (d)
or (d) alone.

Three input files are required to run the program:

Unit 10	Contains the geometric parameter for Rotor 1 together with universal operating conditions (for example temperature, pressure, pitch change axis spacing) and such control switches as are required.
Unit 11	Contains unique geometric parameters for Rotor 2.
Unit 12	Contains specific installation effects input.

All input parameters are documented in the program.

Two files are output:

Unit 16	Contains the output from the "isolated" program.
Unit 14	Contains the output from the installation effects program, including the combined "isolated + installation effects" results if desired.

The output parameters are documented in the program.

CRPCLD (CounterRotation Propeller Coefficients of Lift and Drag) computes blade section lift and drag coefficients as functions of angle–of–attack given counterrotation propeller performance map information as input. This information is used as installation effects input to CRPFAN. Three input files are required to run the program:

Unit 10	Contains geometric parameters for Rotor 1.
Unit 11	Contains geometric parameters for Rotor 2.
Unit 12	Contains performance parameters.

All input parameters are documented in the program.

Two files are output:

Unit 6 Contains the full output in printout form.

Unit 16 contains the calculated coefficients in "NAMELIST" form, suitable for inclusion into the input file for CRPFAN.

The output is documented in the program.

This Volume Contains:

- Acoustic Program
  - CRPFAN Source Code
  - Sample CRPFAN input files:
    - Rotor 1 geometry + operating conditions
    - Rotor 2 geometry
    - Installation Effects
  - Sample CRPFAN output files:
    - Isolated
    - Installation Effects
- Performance Program
  - CRPCLD Source Code
  - Sample CRPCLD input files:
    - Rotor 1 geometry
    - Rotor 2 geometry
    - Performance parameters
  - Sample CRPCLD output files
    - Printout
    - Namelist

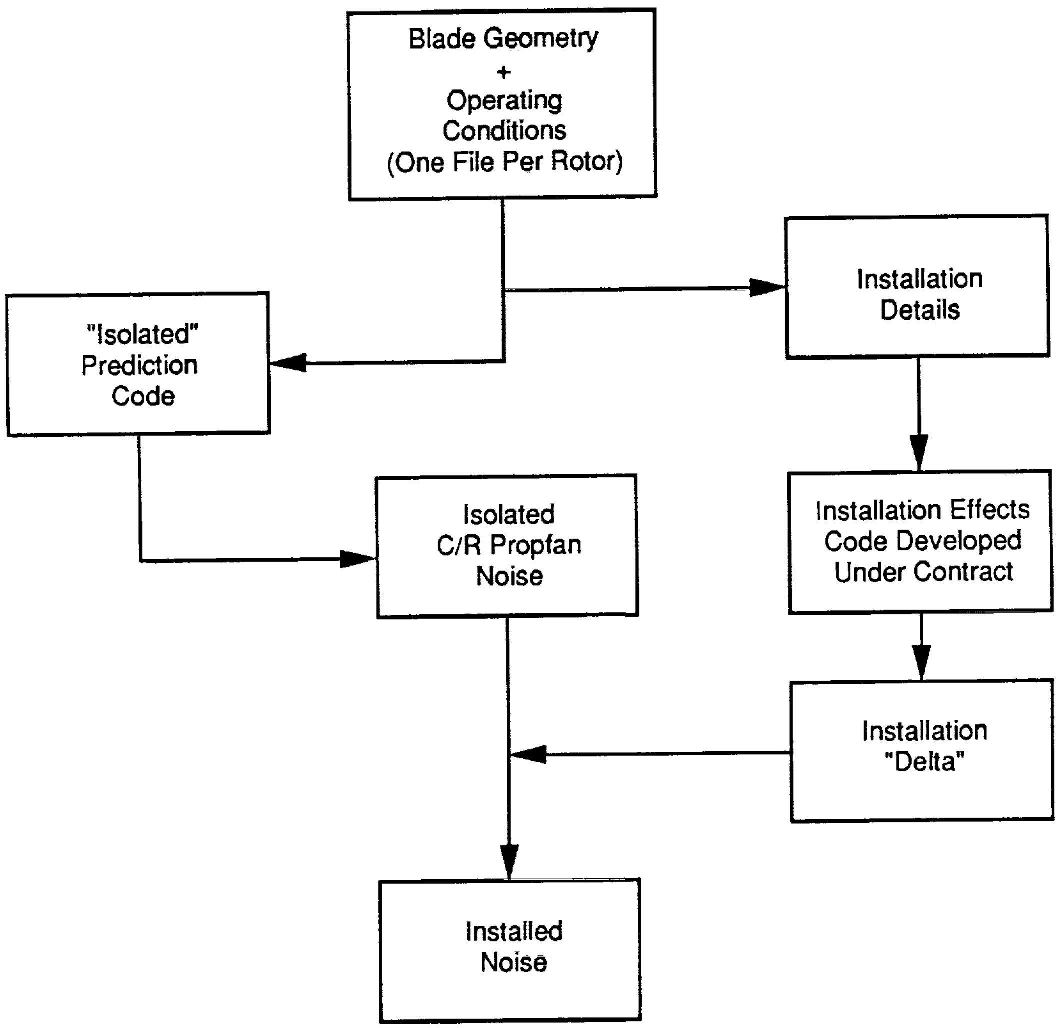

Figure 1. Methodology – Counterrotation Prediction Model.

CRPFAN Program Description

COUNTER–ROTATING PROPFAN NOISE PREDICTION PROGRAM

CRPFAN

The Main Program, CRPFAN, controls program execution. In addition, aerodynamic parameters and uninstalled tone noise are calculated for the two rotors, and printed as follows.

Following the output of input quantities for both rotors (with input YMC & ZMC converted to Face Alignment and Mid–Chord Alignment) Acoustic predictions are output versus observer and emission angles for each rotor.

Tone predictions are identified by frequency and the No. 'M' as follows:

M = 1 –>		NHM:	SL+T harmonics of Rotor 1 BPF
NHM+1 –>		2*NHM:	SL+T harmonics of Rotor 2 BPF
2*NHM+1 –>		NWHM*NHM+2*NHM:	Rotor 1/Rotor 2 UL interaction tones, identified as nnBPF1+mBPF2 where: (M/NHM) = {(nn+1) + (m/NHM)}.

For example: NHM=10 (recommended); NWHM=15:

M = 1 –> 10; 1st 10 harmonics of Rotor 1 BPF
11 –> 20; 1st 10 harmonics of Rotor 2 BPF
21 –> 170; Rotor 1/Rotor 2 interaction tones:
21 = 1F+1A
22 = 1F+2A
23 = 1F+3A ......
121 = 11F+1A ...etc.

Finally, these tones are combined into 1/3 octaves for output.

The following Subroutines are called from CRPFAN:

READIN – Reads input files
GAMCAL – Calculates ratio of specific heats
PITCH1 – Calculates geometric parameters for Rotor 1
LSPFIT – General interpolation and integration routine
PITCH2 – Calculates geometric parameters for Rotor 2
RWVM – Rotor Wake/Vortex Model
SKCAL – Calculates Sears Function (Unsteady Lift Response)
BJFN – Calculates Bessel (J) Functions
AKVCAL – Calculates chordwise non–compactness factor (thickness)
AKLCAL – Calculates chordwise non–compactness factor (loading)
TOBSPL – Combines tones into 1/3 octave bands
ATMOSP – Applies correction for atmospheric absorption
SPLOUT – Prints 1/3 octave output
CRPIE – Counter Rotating Propfan Installation Effects model

SUBROUTINE READIN (Page 42)

This Subroutine reads the input Namelist files containing geometric information for each rotor and program control switches (Namelist 'INPUT'); and, (if the installed calculation is to be performed),installation effects input (Namelist 'INSTAL'). It rearranges the input geometry for input to the installation effects routine (CRPIE) and prints this input.

The following Subroutines are called from READIN:

GAMCAL – Calculates ratio of specific heats

LSPFIT – General interpolation and integration routine

CRPFAN INPUT VARIABLES

Namelist 'INPUT'

Program Control

ISOLAT – Controls Uninstalled C/R Prediction.

=1, Compute "Isolated" Noise, Comprising:

SL+T Noise of Rotor 1;

SL+T Noise of Rotor 2;

UL Noise of Rotor 2 Resulting from Interaction with the Wakes & Vortices shed from Rotor 1.

=0, Do not perform Calculation.

INST – Controls Installed C/R Prediction.

=1, Compute "Installed" Noise resulting from:

Angle–of–Attack effects;

The presence of a Pylon;

The presence of a Cylindrical Fuselage;

The Presence of a Wing Lifting Line.

For Rotor 1 & Rotor 2 harmonics of BPF.

=0, Do not perform Calculation.

NOTE:

If INST=1 & ISOLAT=1 the results from the two predictions are combined.

Definition of "Test" Conditions

NCASE	–	No. of Cases
DT	–	Dt; Tip Diameter, ft.
HTR	–	Hub–to–Tip Radius Ratio.
NB	–	B; No. of Rotor Blades.
SHP	–	Shaft HorsePower.
RPM	–	Rotational Speed, rpm.
DELBP(*)	–	Pitch Angle Change from Design, Delta(Betap) deg.
BETA34(*)	–	Reference Pitch Angle, Beta3/4 deg.
Z34(*)	–	Actual Radius Ratio of Beta3/4.
VO	–	Flight Speed, Vo ft/sec.
TAMB	–	Ambient Temperature To deg. F.
PAMB	–	Ambient Pressure Po psi.
NHM	–	No. of BPF Harmonics to be computed.
NSL	–	Sideline Indicator:
		=1 Sideline prediction
		=0 Constant radius arc prediction.
NTH	–	No. of Observer Angles, Theta–v.
THAD(+)	–	Table (17 max.) of Observer angles, Theta–v deg.
DIST	–	Sideline Distance ro ft.
XPCA	–	Axial Distance between Pitch Change Axes, normalized by Rotor 1 Tip Diameter.

NOTES:

*If BETA34 is included, the value of DELBP in the NAMELIST input is overwritten.
If Z34 is not present, a default value of 0.75 is assumed.

+A default table is present in the program.

Definition of Blading

NZ	–	No. of Spanwise Stations
Z(+)	–	Radius Ratio; z=r/rt
ZMC(+)	–	Axial Location of Blade Mid–Chord Line relative to Pitch–Change Axis, normalized by Tip Diameter, Dt.

YMC(+) – Tangential Location of Blade Mid–Chord Line relative to Pitch–Change Axis, normalized by Tip Diameter Dt.

CHORD(+) – Blade Chord normalized by Tip Diameter, b/Dt.

TMOC(+) – Maximum Blade Section Thickness, normalized by Chord, tm/b.

BETAP(+) – Design or Reference Blade Section Pitch Angle, Betap, deg.

THETA(+) – Blade Section Swirl Coefficient, Theta=(Cu/U)exit.

NOTES:

+NZ required, maximum of 10.

Steady Loading and Thickness Options

ITDO – Thickness Distribution Option Indicator:

=0, Point–Impulsive.
=1, Rectangular.
=2, Parabolic.
=3, Triangular.
=4, Generalized Trapezoidal.
=5, Generalized Parabolic.

ALE++ – Leading Edge Thickness normalized by Max. Thickness, ale/tm.

BTE++ – Trailing Edge Thickness normalized by Max. Thickness, bte/tm.

XTM++ – Position of Max. Thickness relative to Mid–Chord, normalized by Chord.

DXTM – Width of Max. Thickness Region normalized by Chord. (ITDO=4 only).

ILDO – Loading Distribution Option Indicator:

=0, Point–Impulsive.
=1, Rectangular.
=2, Linear.
=3, Generalized Parabolic.

XLM – Position of Max. Loading relative to Mid–Chord, normalized by Chord (ILDO=3 only).

NOTES:

++For ITDO=4 and ITDO=5 only.

General Input

IDS – Doppler Shift Indicator:

=0, No Doppler Shift.
=1, With Doppler Shift.

IPTO – 1/3 – Octave Band Printout Indicator:
=0, No 1/3 – Octave Printout.
=1, With 1/3 – Octave Printout.

ISECT – Interpolated Spanwise Sections Indicator:
=0, Calculations performed using input stations. (Default).
=1, Calculations performed using 51 stations. (Recommended)

ICOSZ – Interpolated Spanwise Sections Distribution Indicator:
=0, Equally Spaced.
=1, Half–Cosine Distribution.

IHEL – Helicoidal Surface Indicator:
=0, Betap Surface used. (Default).
=1, Helicoidal Surface used.

PCTCL – (For short aft blades). Percentage of nominal span removed.

Definition of Near Field Calculation

INF – Near Field Calculation Option Indicator.
=0, Far Field Calculation only. (Default).
=1, Near Field Calculation under Far Field Conditions. (No fuselage or reflecting surfaces present).

INFS – Near Field Calculation Spherical Limit Option.
=0, Limits at Centre of Rotation. (ie; Far Field)
=1, limits at Hub Radius. (Default)

AKINF – Empirical Constant in Near Field Calculation; Default Value = 0.35

Definition of Rotor 1 Wakes/Vortices

IRW – Rotor Wake (i.e., Rotor 2 Unsteady Loading Noise) Calculation Indicator.
=0, No Wake Calculation performed.
=1, Wake Calculation and Unsteady Loading Noise Calculation performed. (Default).

IWAKE – Wake Model Indicator.
=1, Linear Rational Function.
=2, Kemp/Sears. (Default).
=3, Mugridge/Morfey.

ISHAPE – Wake Profile Shape Indicator.
=1, Hyperbolic Secant.
=2, Gaussian. (Default).

ITPVTX – Tip Vortex Option Indicator.
=0, No Tip Vortex Calculation. (Default)
=1, Include Tip Vortex.

IHBVTX(*) – Hub Vortex Option Indicator.
=0, No Hub Vortex Calculation. (Default)
=1, Include Hub Vortex.

IPRNTW – Wake/Vortex Velocity Profile Print Option Indicator.
=0, No Printout. (Default).
=1, Print Profiles.

NWHM – Number of Rotor Wake Gust Harmonics Computed. (Default=Max.=21).

SCD – NZ Values of Drag Coefficient.

BETAW – Parameter to Account for Wake Flow Angle Variation from Freestream. (Default=0.0).

NSTEP – No. of Points in Velocity Profile across one Blade Passage. (Max.=150)

VREF – Reference Velocity for Normalization of Gust Upwash Velocity Harmonic Coefficients, ft/sec. (Default=10.0)

WTIV – Inviscid Velocity Gradient, Normalized by Wheel Speed. (Default=0.0).

CI – Circulation Index for Tip Vortex Strength; (Default=1.0)

SBN(1) – Tangential Distance Normalized by Tip Blade Spacing for the Tip Vortex Center.

SBN(2) – Tangential Distance Normalized by Hub Blade Spacing for the Hub Vortex Center.

TAU – Tip Clearance Normalized by Rotor Tip Chord. (Default=10.0)

TVTI – Tip Vortex Trajectory Index. (Default=2.0)

NOTE:

*The Hub Vortex Option is present in the Code but is not calibrated.

Namelist 'INSTAL'

(This File is only required if INST = 1)

NPCLCD	–	Number of angles (max=51) at which Cl/Cd values are given (same for front and rear rotors) (integer input)
NDES	–	Number of radial spanwise locations at which acoustic calculations are to be performed (Max=51) (integer input)
NRADP	–	Number of radial spanwise locations at which pylon geometry is specified (max=51) (integer input)
THETDP	–	Angular location of pylon: forward looking aft phi (degrees)
CDP	–	Drag coefficient of pylon.
IPYLON	–	IPYLON=0 will cause pylon effect to be not included in acoustic calculations (integer input)
ALD	–	Angles at which front rotor Cl/Cd values are given: NPCLCD values are used (degrees): ALD(1).LT.ALD(2)............LT.ALD(NPCLCD)
CL	–	Front rotor lift coefficient values at angles ALD: NPCLCD values are used.
CD	–	Front rotor drag coefficient values at angles ALD: NPCLCD values are used.
ALDA	–	Angles at which rear rotor Cl/Cd values are given: NPCLCD values are used (degrees): ALDA(1).LT.ALDA(2)............LT.ALDA(NPCLCD)
CLA	–	Rear rotor lift coefficient values at angles ALDA: NPCLCD values are used
CDA	–	Rear rotor drag coefficient values at angles ALDA: NPCLCD values are used
RADP	–	NRADP radial spanwise stations at which pylon geometry is specified, normalized by R1 tip radius: RADP(1).LT.RADP(2)........LT.RADP(NRADP)
XLEP	–	Axial location of pylon leading edge relative to R1 PCA at RADP radii, normalized by R1 tip diameter. +ve upstream; NRADP values used
CAMPD	–	Camber angles of pylon at RADP radii: NRADP values used (degrees)
XTEP	–	Axial location of pylon trailing edge relative to R1 PCA at RADP radii, normalized by R1 tip diameter. +ve upstream; NRADP values used
PSPANV	–	Percent span locations at which acoustic calculations are to be performed: NDES values are used: 0.00.LT.PSPANV.LT.100.00: also PSPANV(1).LT.PSPANV(2) .....LT.PSPANV(NDES)
THFDG	–	(representative thickness)/(max thickness) (used for thickness noise calculations)
CLACC	–	Aircraft wing lift coefficient (0.0 = no wing present)

ARAC – Aircraft wing aspect ratio

AATC – Angle of attack of propellor re: freestream

WSS – Aircraft wing semi–span (ft.)

NPGC – No. of pts. used to eval. Fourier coeff.'s (recommend 32)
The following 3 inputs represent the X,Y,Z location of the Rotor 1 disk center relative to the wing lifting line.

XWL – X dist (ft.) – positive downstream (aft) of lifting line

YWL – Y dist (ft.) – positive horiz. along lifting line to a/c cl

ZWL – Z dist (ft.) – positive vertically above lifting line

ARAD – Acoustic prediction arc radius (ft.) (overwritten by DIST if ISOLAT=1)

NHMAX – Number of bpf harmonics for acoustic predictions (Max=5)

NMODES – Number of modes used in acoustic predictions (Max=10)

NROT – Flag for front propeller rotation direction (FLA) –1 = clockwise1 = counter clockwise

FLEN – Fuselage length (ft.)

FDIA – Max. fuselage dia. (ft.) (0.00 = no fuselage)

FRAWC – Fraction of fuselage length ahead of wing lifting line

NPHLAG – Flag for phase lag in force response calculations
0 = no phase lag
1 = phase lag included

ARBDU – Nondimensional once per rev axial velocity distortion (u/U) : negative on top (PHI= 90 deg.) and positive on bottom (PHI = 270 deg.) for positive ARBDU

The Installation–related input is printed in the installation output file, together with the results of preliminary aerodynamic calculations and the acoustic predictions as follows:

*Preliminary Aerodynamic Output

XV(1) – computed 70 % span pitch (front) after power match
XV(2) – computed 70 % span pitch (rear) after power match
TOTTHF – total thrust (front)
TOTTHR – total thrust (rear)
TOTTH – total thrust
PWRF – power (front)
PWRA – power (rear)
PTOT – total power

*Installation–Related Acoustic Predictions:

I	–	harmonic number index
THTDIV	–	(THAD) input angles for acoustic predictions
PDBSL	–	reference (clean) dB levels to which installation effects delta dB's are applied.
PHDV	–	(PHI) azimuthal angles for acoustic predictions
DIFFDB	–	installation effects delta dB as a function of THTDIV and PHID, relative to reference dB levels at each THTDIV.

SUBROUTINE GAMCAL (Page 54)

Specific heat ratio calculation for air at low pressures from Keenan and Kaye gas tables.

SUBROUTINE PITCH1 (Page 54)

Calculate blade coordinate changes due to pitch angle change relative to 'design' setting —– also calculate Face Alignment (FA) and Mid–Chord Alignment (MCA) —– (Rotor 1)

SUBROUTINE LSPFIT (Page 114)

This routine is a least–squares parabolic curve fit program that will integrate or interpolate using a parabola which passes through points i and i+1 and misses points i–1 and i+2 (if they both exist) such that the square of the deviation is a minimum.

Note that i is generally selected such that;

x(i).le.xc.lt.x(i+1)

the equation for the parabola is:

y–y(i) = b*(x–x(i)) + c*(x–x(i))**2

Outside of the x,y–data range, linear extrapolation of the parabola end point slope is employed if nxtrp = 0.

SUBROUTINE PITCH2 (Page 56)

Calculate blade coordinate changes due to pitch angle change relative to 'design' setting —– also calculate Face Alignment (FA) and Mid–Chord Alignment (MCA) —– (Rotor 2)

SUBROUTINE RWVM (Page 65)

This Subroutine computes the Rotor 1 Wake/Vortex characteristics and evolution with distance downstream. The following Subroutines are called from RWVM:

VORTX1 – Computes strength and radius of tip (and hub) vortices (Page 72)

WAKE1 – Calculates Wake Centerline Defect and Semi–Wake width (Page 70)

WAKE2 – Calculates Tangential Wake Profile (Page 71)

VORTX2 – Computes velocity field induced by vortices at all radii (Page 73)

HRMONIC – Computes harmonic content of Rotor 1 wake/vortex flow at 1/4 chord point of Rotor 2 (Page 77)

These routines are described in NASA CR–1815135, "An Investigation of Counterrotating Tip Vortex Interaction," by R.K. Majjigi, K. Uenishi, and P.R. Gliebe (Reference 7).

SUBROUTINE SKCAL (Page 64)

This subroutine calculates a compressible form of the Sears Function (Unsteady Lift Response). The output is used in the Rotor/Rotor interaction noise calculation.

The following Subroutine is called from SKCAL:

FRESNL – Computes Fresnel Integral Functions C(Z) and S(Z) (Page 63)

SUBROUTINE BJFN (Page 63)

This Subroutine computes J–sub–n(x) Bessel Functions.

SUBROUTINE AKVCAL (Page 59)

This Subroutine calculates the Thickness or Volume Displacement noise source chordwise non–compactness Factor —– k–sub–v —– for the following analytical distributions:

Point–Impulsive

Rectangular

Parabolic

Triangular

Generalized Trapezoidal

Generalized Parabolic

SUBROUTINE AKLCAL (Page 61)

This Subroutine calculates the Steady Loading noise source chordwise non–compactness loading factor —– k–sub–l —– for the following analytical distributions:

Point–Impulsive

Rectangular

Linear

Generalized Parabolic

SUBROUTINE TOBSPL (Page 57)

This Subroutine computes 1/3–octave levels given the tone harmonic spectrum.

SUBROUTINE ATMOSP (Page 62)

This Subroutine applies atmospheric air attenuation corrections for Standard Day (77 deg. F and 70 pct. rel. hum.) conditions from the Shields and Bass correlation to lossless one–third octave spectra.

SUBROUTINE SPLOUT (Page 58)

This Subroutine Calculates dBA and dBD weighted levels, and prints out 1/3–octave SPL tables.

SUBROUTINE CRPIE (Page 80)

This Subroutine calculates 'installation effects' on Counter Rotation Propfan Noise.

It requires access to the 'IMSL' (Edition 10) Library programs: 'CERFE, FFTRF, NEQNF, U4INF, UMINF '.

In addition, the following Subroutines are called from CRPIE or are User–Supplied to the IMSL routines UMINF and NEQNF:

FCNP – Used to find 70% span pitch angles

FCN – Used to find angles of attack on the front and rear Rotors

CLCDEV – Calculates Cl and Cd on front and rear Rotors given alphas

SRPIE – Calculates tone noise deltas from installation effects

SUBROUTINE FCNP (Page 94)

This Subroutine is 'User–supplied' to IMSL solver UMINF. It is used to obtain 70% span pitch angles for the front and rear rotors such that the calculated power absorbed by each rotor approximately equals that input.

The following Subroutines are called from FCNP or are User–Supplied to the IMSL routine NEQNF:

CLCDEV – Calculates Cl and Cd on front and rear Rotors given alphas

FCN – Used to find angles of attack on the front and rear Rotors

SUBROUTINE FCN (Page 92)

This Subroutine is 'User–supplied' to IMSL solver NEQNF. It is used to solve for the steady–state angle of attack (alpha) at each radial station for the front and rear rotors. The following routine is called from FCN:

CLCDEV – Calculates Cl and Cd on front and rear Rotors given alphas

SUBROUTINE CLCDEV (Page 93)

This Subroutine evaluates the Lift and Drag Coefficients on the front and rear rotors given the corresponding angles of attack. The following Subroutine is called from CLCDEV:

LSPFIT – General interpolation and integration routine

SUBROUTINE SRPIE (Page 96)

This Subroutine calculates the effects of the installation environment on the BPF plus harmonics thereof of the forward and aft rotors. The effects are calculated separately for each rotor. The following Subroutines are called from SRPIE:

BJFNA – Calculates Bessel (J) Functions

FCOEFF – Calculates installation–related velocity distortions

PHLAG – Calculates phase lag of force response

OUTPUT – Outputs results of calculation

SUBROUTINE BJFNA (Page 117)

This Subroutine calculates Bessel Functions of the first kind for non–negative integer order and real argument, using the recursion method.

SUBROUTINE FCOEFF (Page 107)

This Subroutine evaluates u,v distortions at the plane of a propeller disk due to angle of attack operation, wing+trailing vortex system, the presence of a mounting pylon upstream, and the presence of a fuselage. It also carries out a Fourier Analysis of these distortions in the 'phi' direction. In addition to the IMSL routine FFTRF, the following Subroutine is called from FCOEFF:

FUSEL – Calculates fuselage induced flowfield

SUBROUTINE FUSEL (Page 111)

This Subroutine computes the parameters needed to evaluate the fuselage induced flow, based on a Rankine Solid model. The IMSL routine NEQNF with the following 'User–Supplied' Subroutine is called from FUSEL:

FCNFU – Function whose zero is to be found for fuselage flow

SUBROUTINE FCNFU (Page 112)

This Subroutine is 'User–supplied' to the IMSL routine NEQNF. It computes the function whose zero is to be found to evaluate the fuselage induced flow based on a Rankine Solid model.

SUBROUTINE PHLAG (Page 113)

This Subroutine computes two dimensional, linearised, compressible, flat plate convected gust phase lag formulae based on Amiet's low frequency theory where applicable and on a Fresnel Integral based formula otherwise. See Goldstein: 'Aeroacoustics': pages 138–139, eqs.(3.70) and (3.71) for these formulae. In addition to the IMSL routine CERFE, the following Subroutine is called from PHLAG:

BJFNA – Calculates Bessel (J) Functions (Page 117)

SUBROUTINE OUTPUT (Page 106)

This Subroutine writes out the Final Results of the predicted effects of installation environment. These Results are represented acoustically by Delta dBs and are presented in table form for each harmonic of BPF as Delta dB vs. angle Theta and angle Phi. Results for the front rotor are written first, followed by the results for the aft rotor. The Final set of information combines the installation effect Deltas calculated in CRPIE with the BPF harmonic tones predicted in the CRPFAN main program.

CRPFAN Source Code

```
C ***************************************************************
C
C ****  COUNTER-ROTATING PROPFAN NOISE PREDICTION PROGRAM  ****
C
C ****                 C R P F A N                         ****
C
C ***************************************************************
C
C ***************************************************************
C *                CRPFAN INPUT VARIABLES                       *
C ***************************************************************
C ***************************************************************
C
C              CRPFILE INPUT (UNITS 10 & 11)
C
C                         Program Control
C
C  ISOLAT         -         Controls Uninstalled C/R Prediction.
C                            =1, Compute "Isolated" Noise, Comprising:
C                                 SL+T Noise of Rotor 1;
C                                 SL+T Noise of Rotor 2;
C                                 UL Noise of Rotor 2 Resulting from Interaction
C                                 with the Wakes & Vortices shed from Rotor 1.
C                            =0, Do not perform Calculation.
C
C  INST           -         Controls Installed C/R Prediction.
C                            =1, Compute "Installed" Noise resulting from:
C                                 Angle-of-Attack effects;
C                                 The presence of a Pylon;
C                                 The presence of a Cylindrical Fuselage;
C                                 The Presence of a Wing Lifting Line.
C                                For Rotor 1 & Rotor 2 harmonics of BPF.
C                            =0, Do not perform Calculation.
C
C
C  Note:
C      If INST=1 & ISOLAT=1 the results from the two predictions are combined.
C
C
C ******************************************************************
C
C                    Definition of "Test" Conditions
C
C
C  NCASE          -         No. of Cases
C
C  DT             -         Dt; Tip Diameter, ft.
C
C  HTR            -         Hub-to-Tip Radius Ratio.
C
C  NB             -         B; No. of Rotor Blades.
C
C  SHP            -         Shaft HorsePower.
C
C  RPM            -         Rotational Speed, rpm.
C
C  DELBP(*)       -         Pitch Angle Change from Design, Delta(Betap) deg.
C
```

```
C  BETA34(*)      -        Reference Pitch Angle, Beta3/4 deg.
C
C  Z34(*)         -        Actual Radius Ratio of Beta3/4.
C
C  VO             -        Flight Speed, Vo ft/sec.
C
C  TAMB           -        Ambient Temperature To deg. F.
C
C  PAMB           -        Ambient Pressure Po psi.
C
C  NHM            -        No. of BPF Harmonics to be computed.
C
C  NSL            -        Sideline Indicator:
C                           =1 Sideline prediction
C                           =0 Constant radius arc prediction.
C
C  NTH            -        No. of Observer Angles, Theta-v.
C
C  THAD(+)        -        Table (17 max.) of Observer angles, Theta-v deg.
C
C  DIST           -        Sideline Distance ro ft.
C
C  XPCA           -        Axial Distance between Pitch Change Axes,
C                          normalised by Rotor 1 Tip Diameter.
C
C
C  Notes:
C  *   If BETA34 is included, the value of DELBP in the NAMELIST input is
C      overwritten.
C      If Z34 is not present, a default value of 0.75 is assumed.
C  +   A default table is present in the program.
C
C
C *******************************************************************
C
C                         Definition of Blading
C
C
C  NZ             -        No. of Spanwise Stations
C
C  Z(+)           -        Radius Ratio; z=r/rt
C
C  ZMC(+)         -        Axial Location of Blade Mid-Chord Line relative
C                          to Pitch-Change Axis, normalised by Tip
C                          Diameter, Dt.
C
C  YMC(+)         -        Tangential Location of Blade Mid-Chord Line
C                          relative to Pitch-Change Axis, normalised by
C                          Tip Diameter Dt.
C
C  CHORD(+)       -        Blade Chord normalised by Tip Diameter, b/Dt.
C
C  TMOC(+)        -        Maximum Blade Section Thickness, normalised by
C                          Chord, tm/b.
C
C  BETAP(+)       -        Design or Reference Blade Section Pitch Angle,
C                          Betap, deg.
C
```

```
C  THETA(+)       -         Blade Section Swirl Coefficient, Theta=(Cu/U)exit.
C
C
C  Notes:
C  +   NZ required, maximum of 10.
C
C
C ******************************************************************
C
C                 Steady Loading and Thickness Options
C
C
C  ITDO           -         Thickness Distribution Option Indicator:
C                            =0, Point-Impulsive.
C                            =1, Rectangular.
C                            =2, Parabolic.
C                            =3, Triangular.
C                            =4, Generalised Trapezoidal.
C                            =5, Generalised Parabolic.
C
C  ALE++          -         Leading Edge Thickness normalised by Max.
C                           Thickness, ale/tm.
C
C  BTE++          -         Trailing Edge Thickness normalised by Max.
C                           Thickness, bte/tm.
C
C  XTM++          -         Position of Max. Thickness relative to
C                           Mid-Chord, normalised by Chord.
C
C  DXTM           -         Width of Max. Thickness Region normalised by
C                           Chord. (ITDO=4 only).
C
C  ILDO           -         Loading Distribution Option Indicator:
C                            =0, Point-Impulsive.
C                            =1, Rectangular.
C                            =2, Linear.
C                            =3, Generalised Parabolic.
C
C  XLM            -         Position of Max. Loading relative to Mid-Chord,
C                           normalised by Chord. (ILDO=3 only).
C
C
C  Notes:
C  ++   For ITDO=4 and ITDO=5 only.
C
C
C ******************************************************************
C
C                          General Input
C
C
C  IDS            -         Doppler Shift Indicator:
C                            =0, No Doppler Shift.
C                            =1, With Doppler Shift.
C
C  IPTO           -         1/3 - Octave Band Printout Indicator:
C                            =0, No 1/3 - Octave Printout.
C                            =1, With 1/3 - Octave Printout.
```

```
C
C  ISECT          -        Interpolated Spanwise Sections Indicator:
C                           =0, Calculations performed using input
C                             stations. (Default).
C                           =1, Calculations performed using 51 stations.
C                             (Recommended)
C
C  ICOSZ          -        Interpolated Spanwise Sections Distribution
C                          Indicator:
C                           =0, Equally Spaced.
C                           =1, Half-Cosine Distribution.
C
C  IHEL           -        Helicoidal Surface Indicator:
C                           =0, Betap Surface used. (Default).
C                           =1, Helicoidal Surface used.
C
C  PCTCL          -        (For short aft blades). Percentage of nominal
C                          span removed.
C
C
C ******************************************************************
C
C                 Definition of Near Field Calculation
C
C
C  INF            -        Near Field Calculation Option Indicator.
C                           =0, Far Field Calculation only. (Default).
C                           =1, Near Field Calculation under Far Field
C                             Conditions. (No fuselage or reflecting
C                             surfaces present).
C
C                           =2, Near Field Calculation on Exterior of
C                             Fuselage; includes reflection and
C                             refraction and boundary layer broadband
C                             noise.
C                           =3, Near Field Calculation Inside Cabin;
C                             includes all effects in INF=2 option plus
C                             cabin wall transmission loss effects.
C
C  INFS           -        Near Field Calculation Spherical Limit Option.
C                           =0, Limits at Centre of Rotation. (ie Far Field)
C                           =1, Limits at Hub Radius. (Default)
C
C  AKINF          -        Empirical Constant in Near Field Calculation;
C                          Default Value = 0.35
C
C
C ******************************************************************
C
C                 Definition of Rotor 1 Wakes/Vortices
C
C
C  IRW            -        Rotor Wake (ie Rotor 2 Unsteady Loading Noise)
C                          Calculation Indicator.
C                           =0, No Wake Calculation performed.
C                           =1, Wake Calculation and Unsteady Loading Noise
C                             Calculation performed. (Default).
C
```

```
C  IWAKE          -          Wake Model Indicator.
C                             =1, Linear Rational Function.
C                             =2, Kemp/Sears. (Default).
C                             =3, Mugridge/Morfey.
C
C  ISHAPE         -          Wake Profile Shape Indicator.
C                             =1, Hyperbolic Secant.
C                             =2, Gaussian. (Default).
C
C  ITPVTX         -          Tip Vortex Option Indicator.
C                             =0, No Tip Vortex Calculation. (Default)
C                             =1, Include Tip Vortex.
C
C  IHBVTX(*)      -          Hub Vortex Option Indicator.
C                             =0, No Hub Vortex Calculation. (Default)
C                             =1, Include Hub Vortex.
C
C  IPRNTW         -          Wake/Vortex Velocity Profile Print Option
C                            Indicator.
C                             =0, No Printout. (Default).
C                             =1, Print Profiles.
C
C  NWHM           -          Number of Rotor Wake Gust Harmonics Computed.
C                            (Default=Max.=21).
C
C  SCD            -          NZ Values of Drag Coefficient.
C
C  BETAW          -          Parameter to Account for Wake Flow Angle Variation
C                            from Freestream. (Default=0.0).
C
C  NSTEP          -          No. of Points in Velocity Profile across one Blade
C                            Passage. (Max.=150)
C
C  VREF           -          Reference Velocity for Normalisation of Gust Upwash
C                            Velocity Harmonic Coefficients, ft/sec.
C                            (Default=10.0)
C
C  WTIV           -          Inviscid Velocity Gradient, Normalised by Wheel
C                            Speed. (Default=0.0).
C
C  CI             -          Circulation Index for Tip Vortex Strength;
C                            (Default=1.0)
C
C  SBN(1)         -          Tangential Distance Normalized by Tip Blade Spacing
C                            for the Tip Vortex Center.
C
C  SBN(2)         -          Tangential Distance Normalized by Hub Blade Spacing
C                            for the Hub Vortex Center.
C
C  TAU            -          Tip Clearance Normalized by Rotor Tip Chord.
C                            (Default=10.0)
C
C  TVTI           -          Tip Vortex Trajectory Index.
C                            (Default=2.0)
C
C  Note:
C  *   The Hub Vortex Option is present in the Code but is not calibrated.
C
```

```
C *******************************************************************
C
C              'ISOLATED' OUTPUT (UNIT 16)
C
C  Following the output of input quantities for both rotors (with
C  input YMC & ZMC converted to Face Alignment and Mid-Chord Alignment)
C  Acoustic predictions are output versus observer and emission angles
C  for each rotor.
C
C  Tone predictions are identified by frequency and the No. 'M' as
C  follows:
C          M =        1 ->             NHM:  SL+T harmonics of Rotor 1 BPF
C               NHM+1 ->           2*NHM:  SL+T harmonics of Rotor 2 BPF
C             2*NHM+1 -> NWHM*NHM+2*NHM:  Rotor 1/Rotor 2 UL interaction tones,
C                                          identified as nnBPF1+mBPF2 where:
C                                          (M/NHM) = {(nn+1) + (m/NHM)}.
C
C     For example: NHM=10 (recommended); NWHM=15:
C
C           M =  1 ->  10; 1st 10 harmonics of Rotor 1 BPF
C               11 ->  20; 1st 10 harmonics of Rotor 2 BPF
C               21 -> 170; Rotor 1/Rotor 2 interaction tones:
C                           21 =  1F+1A
C                           22 =  1F+2A
C                           23 =  1F+3A ......
C                          121 = 11F+1A ...etc.
C
C
C  Finally, these tones are combined into 1/3 octaves for output.
C
C *******************************************************************
C *******************************************************************
C
C                CRPIE INPUT (UNIT 12)
C
C       (This File is only required if INSTAL = 1)
C
C
C     NPCLCD   - number of angles (max=51) at which Cl/Cd values
C                are given ( same for front and rear rotors )
C                            ( integer input )
C
C     NDES     - number of radial spanwise locations at which
C                  acoustic calculations are to be performed
C                   (Max=51) ( integer input )
C
C     NRADP    - number of radial spanwise locations at which
C                 pylon geometry is specified (max=51)
C                            ( integer input )
C
C     THETDP   - angular location of pylon : forward looking
C                  aft phi ( degrees )
C
C     CDP      - drag coefficient of pylon
C
C     IPYLON   - IPYLON=0 will cause pylon effect to be not
C                 included in acoustic calculations
C                            ( integer input )
```

```
C
C     ALD      - angles at which front rotor Cl/Cd values are
C                  given : NPCLCD values are used ( degrees ):
C                 ALD(1).LT.ALD(2)...........LT.ALD(NPCLCD)
C
C     CL       - front rotor lift coefficient values at angles
C                    ALD : NPCLCD values are used
C
C     CD       - front rotor drag coefficient values at angles
C                    ALD : NPCLCD values are used
C
C     ALDA     - angles at which rear rotor Cl/Cd values are
C                  given : NPCLCD values are used ( degrees ):
C                 ALDA(1).LT.ALDA(2)...........LT.ALDA(NPCLCD)
C
C     CLA      - rear rotor lift coefficient values at angles
C                    ALDA : NPCLCD values are used
C
C     CDA      - rear rotor drag coefficient values at angles
C                    ALDA : NPCLCD values are used
C
C     RADP     - NRADP radial spanwise stations at which pylon
C                 geometry is specified, normalized by R1 tip radius.
C                 RADP(1).LT.RADP(2)........LT.RADP(NRADP)
C
C     XLEP     - axial location of pylon leading edge relative to R1 PCA
C                 at RADP radii, normalized by R1 tip diameter.
C                  +ve upstream; NRADP values used
C
C     CAMPD    - camber angles of pylon at RADP radii:
C                   NRADP values used ( degrees )
C
C     XTEP     - axial location of pylon trailing edge relative to R1 PCA
C                 at RADP radii, normalized by R1 tip diameter.
C                  +ve upstream; NRADP values used
C
C     PSPANV   - per cent span locations at which acoustic
C                  calculations are to be performed: NDES
C                  values are used : 0.00.LT.PSPANV.LT.100.00:
C                  also PSPANV(1).LT.PSPANV(2).....LT.PSPANV(NDES)
C
C     THFDG    - (representative thickness)/(max thickness)
C                 (used for thickness noise calculations)
C
C     CLACC    - aircraft wing lift coefficient (0.0 = no wing present)
C
C     ARAC     - aircraft wing aspect ratio
C
C     AATC     - angle of attack of propellor re: freestream
C
C     WSS      - aircraft wing semi-span (ft.)
C
C     NPGC     - no. of pts. used to eval. Fourier coeff.'s (recommend 32)
C
C   the following 3 inputs represent the X,Y,Z location of the
C   Rotor 1 disk center relative to the wing lifting line.
C
C     XWL      - X dist (ft.) - positive downstream (aft) of lifting line
```

```
C      YWL      - Y dist (ft.) - positive horiz. along lifting line to a/c cl
C      ZWL      - Z dist (ft.) - positive vertically above lifting line
C
C      ARAD     - acoustic prediction arc radius (ft.)
C                  (overwritten by DIST if ISOLAT=1)
C
C      NHMAX    - number of bpf harmonics for acoustic predictions (Max=5)
C
C      NMODES   - number of modes used in acoustic predictions (Max=10)
C
C      NROT     - flag for front propeller rotation direction (FLA)
C                 -1 = clockwise
C                  1 = counter clockwise
C
C      FLEN     - fuselage length (ft.)
C
C      FDIA     - max. fuselage dia. (ft.) (0.00 = no fuselage)
C
C      FRAWC    - fraction of fuselage length ahead of wing lifting line
C
C      NPHLAG   - flag for phase lag in force response calculations -
C                 0 = no phase lag
C                 1 = phase lag included
C
C      ARBDU    - non dimensional once per rev axial velocity
C                 distortion ( u/U ) : negative on top ( PHI= 90 deg.)
C                 and positive on bottom ( PHI = 270 deg.) for
C                 positive ARBDU
C
C * OUTPUT OF INSTALLATION-RELATED INPUTS: (UNIT 14)
C
C      RHO,TAMB,PO,GAMC,MFLT,ASEPIN,ARBDU,AATC,THFDG,CDP,THETDP,
C      RADP,XLEP,XTEP,CAMPD,(message if IPYLON=0),RPM,RTIPF,NBF,BET701,
C      PFRONT,XPCHA,NROT,ALD,CL,CD,RADV,CHORD,PITCH,YMC,ZMC,TMOC,PSPANV,
C      RPMA,RTIPA,NBA,BET702,PREAR,XPCHAR,NROT related for rear rotor,
C      ALDA,CLA,CDA,RADVA,CHORDA,PITCHA,YMCA,ZMCA,TMOC2,CLACC,ARAC,WSS,
C      FLEN,FDIA,FRAWC,XWL,YWL,ZWL
C
C * OUTPUT OF SOME INSTALLATION-RELATED CALCULATED QUANTITIES
C
C      XV(1)  - computed 70 % span pitch ( front ) after power match
C      XV(2)  - computed 70 % span pitch (  rear ) after power match
C      TOTTHF - total thrust ( front )
C      TOTTHR - total thrust (  rear )
C      TOTTH  - total thrust
C      PWRF   - power ( front )
C      PWRA   - power (  rear )
C      PTOT   - total power
C
C * OUTPUT OF INSTALLATION EFFECT PREDICTION RESULTS FOR EACH
C   BPF HARMONIC NUMBER (FROM 1 TO NHMAX) FOR EACH ROTOR :
C
C      I      - harmonic number index
C      THTDIV - (THAD) input angles for acoustic predictions
C      PDBSL  - reference (clean) dB levels to which installation
C               effects delta dB's are applied.
C      PHDV   - (PHI) azimuthal angles for acoustic predictions
C      DIFFDB - installation effects delta dB as a function of THTDIV
```

```
C               and PHID, relative to reference dB levels at each THTDIV.
C
C *********************************************************************
C *
C *                FINAL CONSOLIDATED OUTPUT
C *
C       Applies installation Effects Delta dBs to BPF harmonic tone
C       levels calculated by 'isolated' model.
C *
C *********************************************************************
C
        COMMON /COM12/  NZUSE,DTR,RTD,VO,CO,SXOCH(51)
C
        COMMON /NLCOM/
     & AKINF,ALE(10),BETA34,BETAP(10),BTE(10),CHORD(10),
     & DELBP,DIST,DT,DXTM,HTR,ICOSZ,IDS,IHEL,ILDO,INF,INFS,
     & INST,IPTO,IRW,ISECT,ISOLAT,ITDO,NALE,NB,NBTE,NCASE,NHM,
     & NP,NSL,NSTEP,NTH,NRSEC,NZ,PAMB,PCTCL,RPM,RSEC(51),SCD(10),
     & SHP,SIGR,TAMB,THETA(10),THAD(17),TMOC(10),WTIV,XLM,
     & XPCA,XTM,YMC(10),Z(10),ZMC(10),Z34
C
        COMMON /INCOM/
     & NZ11,B341,DT1,NHM1,RPM1,SHP1,HTR1,ITDO1,XTM1,DXTM1,
     & YOD1,XLM1,ALE1(51),ALE11(10),BP11(10),BTE1(51),BTE11(10),
     & CHD11(10),ETA1,RSEC1(51),THAD1(17),TMOC11(10),YMC11(10),
     & Z11(10),ZMC11(10),ILDO1,
     & NZ21,B342,DT2,NHM2,RPM2,SHP2,HTR2,ITDO2,XTM2,DXTM2,
     & YOD2,XLM2,ALE2(51),ALE21(10),BP21(10),BTE2(51),BTE21(10),
     & CHD21(10),ETA2,RSEC2(51),THAD2(17),TMOC21(10),YMC21(10),
     & Z21(10),ZMC21(10),ILDO2
C
        COMMON /RWAKE1/ ISHAPE,BETAW,NWHM,VREF,IWAKE
C
        COMMON /VTEX1/ ITPVTX,IHBVTX,TAU,ALPHR,DUMMY,WT00,B1
C
        COMMON /VTEX2/ SAODS(2),SCIRC0(2),SO0(2),SVSDV0(2),SCL(2),
     &                 SFRL(2)
C
        COMMON /VTEX3/ SBN(2),SBR(2),SDIST,RAWDS,R,VINRVT(300,2),
     &                 VISRVT(300,2),VHTR,VINRV(300),VISRV(300)
C
        COMMON /VTEX4/ CLAV,CAV,WT00T,ABR(51),CI,TVTI
C
        COMMON /HMONIC/ ST(300),VPN(300),FCA(21),FCB(21),FCDB(21)
C
        COMMON /PRINTD/ IPRT,IPR,IPRNTW
C
        COMMON /BLADE1/ THETA1(51), SIGR1(51), DPINT1(51), CHORD1(51),
     & TMOC1(51), Z1(51), ZMC1(51), YMC1(51), BETAP1(51), MCA1(51),
     & FA1(51), BETPP1(51), DELBP1, SCD1(51), EMT1, NB1
C
        COMMON /BLADE2/ THETA2(51), SIGR2(51) ,DPINT2(51), CHORD2(51),
     & TMOC2(51), Z2(51), ZMC2(51), YMC2(51), BETAP2(51), MCA2(51),
     & FA2(51), BETPP2(51), DELBP2, SCD2(51), EMT2, NB2
C
        COMMON /CRINST/ BPHM(5,17,2)
C
        REAL MCA,MCA1,MCA2
```

```
C
        DIMENSION SPLT(17,230), FREQ(17,230), SPL(17,27), IFREQ(27)
        DIMENSION CL1(51),EMR1(51),DPMBR(51,230),DPMBI(51,230)
        DIMENSION CL2(51),EMR2(51)
        DIMENSION XC1(51),YC1(51),XI(51),YI(51),XC2(51),YC2(51)
        DIMENSION THE1(17),SNTHE1(17),CSTHE1(17),CVF1(17),THED1(17)
        DIMENSION THE2(17),SNTHE2(17),CSTHE2(17),CVF2(17),THED2(17)
        DIMENSION THED(17),EMRR(51),SPARE(51),RCLQ(51,21),RPHIG(51,21)
C
C *** SET PROGRAM DEFAULT VALUES
C
        DATA NHM,NTH,NSL,DIST,NWHM /25,17,0,100.0,15/
        DATA THAD /10.0,20.0,30.0,40.0,50.0,60.0,70.0,80.0,90.0,
      & 100.0,110.0,120.0,130.0,140.0,150.0,160.0,170.0/
        DATA RG /1716.26/
        DATA IPTO,IDS,ITDO1,ITDO2 /0,0,2,2/
        DATA ILDO1,ILDO2,INF,INFS /2,2,0,1/
        DATA IFREQ /50,63,80,100,125,160,200,250,315,
      & 400,500,630,800,1000,1250,1600,2000,2500,3150,4000,
      & 5000,6300,8000,10000,12500,16000,20000/
        DATA WTIV     /0./
        DATA ICOSZ,NP /1,1/
        DATA AKINF    /0.35/
        DATA IWAKE,IRW,IQMAXR/2,1,10/
C
C      IWAKE=1 LINEAR RATIONAL FUNCTION MODEL FOR ROTOR WAKE
C      IWAKE=2 SILVERSTEIN/KEMP & SEARS MODEL FOR ROTOR WAKE(DEFAULT)
C      IWAKE=3 MUGRIDGE & MORFEY MODEL FOR ROTOR WAKE
C
        DATA WTIV,PCTCL /0.,0./
        DATA ITPVTX,IHBVTX,PCTVTX,CI,TAU,TVTI/0,0,30.,1.,10.,2./
C
C       PCTVTX=% OUTBOARD SPAN ROLLED UP INTO TIP VORTEX
C       CI= CIRCULATION INDEX FOR TIP VORTEX
C       TAU=EFFECTIVE TIP CLEARANCE/ROTOR TIP CHORD TO SIMULATE IMAGE VTX
C       TVTI=TIP VORTEX TRAJECTORY INDEX
C
C *** INITIALIZE PROGRAM CONSTANTS
C
        PI   = 3.1415926
        TPI  = 2.0*PI
        DTR  = PI/180.0
        RTD  = 1.0/DTR
C
 100   CONTINUE
C
        DO 110 M = 1,230
          DO 110 IZ = 1,51
            DPMBR(IZ,M) = 0.
            DPMBI(IZ,M) = 0.
 110   CONTINUE
C
C *** READ INPUT AND SET UP GEOMETRY FOR BOTH ROTOR SYSTEMS
C       IF ONLY INSTALLATION EFFECTS WANTED, SKIP TO
C                       CRPIE CALL
C
        CALL READIN
C
```

```
      IF(ISOLAT.EQ.0) GO TO 700
C
C ***********************************************************************
C *** START MAIN PROGRAM CALCULATIONS ***********************************
C ***********************************************************************
C
      WRITE (16,1500)
C
      IF(IHEL .NE. 1) WRITE (16,1540)
      IF(IHEL .EQ. 1) WRITE (16,1560)
C
      WRITE (16,1600)
      WRITE (16,1640) DT1,NB1,SHP1,RPM1,VO,TAMB,PAMB,NP
      WRITE (16,1620)
      WRITE (16,1640) DT2,NB2,SHP2,RPM2,VO,TAMB,PAMB,NP
C
      WRITE (16,1660) XPCA,IRW,IWAKE,IDS,INF,INFS
C
C
C *** PRELIMINARY AERODYNAMIC PARAMETER CALCULATIONS
C
C
      RPS1 = RPM1/60.0
      B1   = NB1
      UT1  = PI*DT1*RPS1
      TO   = TAMB+459.7
      PO   = 144.0*PAMB
      RHO  = PO/(RG*TO)
C
      CALL GAMCAL(TAMB,GAMMA)
C
      CO    = SQRT(GAMMA*RG*TO)
      EMT1  = UT1/CO
      EMO   = VO/CO
      EMH1  = SQRT(EMT1**2+EMO**2)
      WCON1 = 0.25*GAMMA*PO*CO*EMO*EMT1*B1*DT1*DT1
      PCON1 = GAMMA*PO*B1/(4.0*PI*SQRT(2.0))
      QO    = 0.5*GAMMA*PO*EMO*EMO
      ADVR1 = VO/(RPS1*DT1)
      WRITE (16,1680) EMT1,EMO,EMH1
C
      RPS2  = RPM2/60.0
      B2    = NB2
      UT2   = PI*DT2*RPS2
      EMT2  = UT2/CO
      EMH2  = SQRT(EMT2**2+EMO**2)
      WCON2 = 0.25*GAMMA*PO*CO*EMO*EMT2*B2*DT2*DT2
      PCON2 = GAMMA*PO*B2/(4.0*PI*SQRT(2.0))
      ADVR2 = VO/(RPS2*DT2)
      WRITE(16,1680) EMT2,EMO,EMH2
C
C *** BLADE SECTION PROPERTIES CALCULATION - ROTOR 1
C
      CALL PITCH1
C
      THMOD1=1.0
C
 120  CONTINUE
```

```
C
         DO 130 IZ = 1,NZUSE
           SIGR1(IZ)   = B1*CHORD1(IZ)/(PI*Z1(IZ))
           EMR1(IZ)    = SQRT(EMO**2+(EMT1*Z1(IZ))**2)
           CLCON = 2.0/(GAMMA*EMR1(IZ)*EMR1(IZ))
           DPINT1(IZ) = GAMMA*EMT1*Z1(IZ)*EMR1(IZ)*THETA1(IZ)/SIGR1(IZ)
           DPINT1(IZ) = THMOD1*DPINT1(IZ)
           CL1(IZ)     = CLCON*DPINT1(IZ)
           YC1(IZ)     = EMR1(IZ)*CL1(IZ)*CHORD1(IZ)*Z1(IZ)
 130    CONTINUE
C
         IZVTX = 1
         ZVTX  = Z1(1)-PCTVTX*(Z1(1)-HTR1)/100.
C
         DO 140 IZ = 1,NZUSE
           IF(Z1(IZ) .GT. ZVTX) GO TO 140
             IZVTX = IZ
             GO TO 150
 140     CONTINUE
C
 150     CONTINUE
C
         SCLAV = 0.0
         SCAV  = 0.0
C
         DO 160 IZ = 1,IZVTX
           SCLAV = SCLAV+CL1(IZ)
           SCAV  = SCAV+CHORD1(IZ)
 160    CONTINUE
C
         FIZVTX = FLOAT(IZVTX)
         CLAV   = SCLAV/FIZVTX
         CAV    = SCAV/FIZVTX
C
C *** ROTOR EFFECTIVE AERODYNAMIC HORSEPOWER AND DISK LOADING
C
         XI(1) = HTR1
         XI(2) = 1.0
         YI(1) = 0.0
         IF(Z1(1) .LT. Z1(NZUSE)) GO TO 170
         XI(1) = 1.0
         XI(2) = HTR1
         YI(1) = 0.0
 170    CONTINUE
C
         CALL LSPFIT(Z1,YC1,NZUSE,XI,YI,2,-1)
C
         IF(Z1(1) .GT. Z1(NZUSE)) YI(2) = -YI(2)
         SHPE1  = WCON1*YI(2)/550.0
         IF(ABS(SHP1-SHPE1) .LE. 0.01) GO TO 180
         THMOD1 = SHP1/SHPE1
         GO TO 120
C
C *** PRINT-OUT BLADE SECTION PROPERTIES - ROTOR 1
C
 180    CONTINUE
C
         WRITE(16,1840)
```

```
        IPRT = 1
        IF(NZUSE .GT. 15) IPRT = NZUSE/10
C
        DO 190 IZ = 1,NZUSE,IPRT
          WRITE(16,1880)IZ,Z1(IZ),CHORD1(IZ),TMOC1(IZ),
     &  BETPP1(IZ),MCA1(IZ),FA1(IZ),CL1(IZ),SCD1(IZ)
 190  CONTINUE
C
C *** BLADE SECTION PROPERTIES CALCULATION - ROTOR 2
C
        CALL PITCH2
C
        THMOD2 = 1.0
C
        IZCL = 1
        IF (PCTCL.GT.0.) THEN
            ZCL = Z2(1)-PCTCL*(Z2(1)-HTR2)/100.
            DO 200 IZ=1,NZUSE
                IF(Z2(IZ).LE.ZCL) THEN
                    IZCL = IZ
                    GO TO 205
                END IF
 200        CONTINUE
        END IF
 205    CONTINUE
C
        DO 210 IZ = 1,NZUSE
          SIGR2(IZ) = B2*CHORD2(IZ)/(PI*Z2(IZ))
          EMRR(IZ)  = SQRT(EMO**2+(Z2(IZ)*EMT2+THETA1(IZ)*THMOD1
     &                  *Z1(IZ)*EMT1)**2)
          EMR2(IZ)  = EMRR(IZ)
          CLCON     = 2.0/(GAMMA*EMR2(IZ)*EMR2(IZ))
          DPINT2(IZ)= GAMMA*EMT2*Z2(IZ)*EMR2(IZ)*THETA2(IZ)/SIGR2(IZ)
          DPINT2(IZ)= THMOD2*DPINT2(IZ)
          CL2(IZ)   = CLCON*DPINT2(IZ)
          YC2(IZ)   = EMRR(IZ)*CL2(IZ)*CHORD2(IZ)*Z2(IZ)
 210  CONTINUE
C
C *** ROTOR EFFECTIVE AERODYNAMIC HORSEPOWER AND DISK LOADING
C
        XI(1) = HTR2
        XI(2) = Z2(IZCL)
        YI(1) = 0.0
        NZCL  = NZUSE-IZCL+1
        IF(Z2(1) .LT. Z2(NZUSE)) GO TO 220
          XI(1) = Z2(IZCL)
          XI(2) = HTR2
          YI(1) = 0.0
 220  CONTINUE
C
        CALL LSPFIT(Z2(IZCL),YC2(IZCL),NZCL, XI,YI,2, -1)
C
        IF(Z2(1) .GT. Z2(NZUSE)) YI(2) = -YI(2)
        SHPE2  = WCON2*YI(2)/550.0
        IF(ABS(SHP2-SHPE2) .LE. 0.01) GO TO 230
        THMOD2 = SHP2/SHPE2
        GO TO 205
C
```

```
C *** PRINT-OUT BLADE SECTION PROPERTIES - ROTOR 2
C
 230  CONTINUE
        WRITE(16,1980)
        DO 240 IZ = IZCL,NZUSE,IPRT
          WRITE(16,1880)IZ,Z2(IZ),CHORD2(IZ),TMOC2(IZ),
     &  BETPP2(IZ),MCA2(IZ),FA2(IZ),CL2(IZ)
 240  CONTINUE
C
C *** ROTOR 1
C
        AA1  = 0.25*PI*DT1*DT1*(1.0-HTR1*HTR1)
        DL11 = SHPE1/(DT1*DT1)
        DL21 = SHPE1/AA1
        CPQ1 = 550.0*SHPE1/(RHO*(RPS1**3)*(DT1**5))
        WRITE(16,2020)
        WRITE(16,2040) SHPE1,AA1,DL11,DL21,CPQ1,UT1,B341
C
C *** ROTOR 2
C
        AA2  = 0.25*PI*DT2*DT2*(1.0-HTR2*HTR2)
        DL12 = SHPE2/(DT2*DT2)
        DL22 = SHPE2/AA2
        CPQ2 = 550.0*SHPE2/(RHO*(RPS2**3)*(DT2**5))
        WRITE(16,2060)
        WRITE(16,2040) SHPE2,AA2,DL12,DL22,CPQ2,UT2,B342
C
C **** CALCULATE EMISSION ANGLES AND FUNCTIONS OF SAME
C
C *** ROTOR 1
C
        DO 250 J = 1,NTH
          SINX = EMO*SIN(DTR*THAD1(J))
          COSX = SQRT(1.0-SINX*SINX)
          TANX = SINX/COSX
          THE1(J)    = DTR*THAD1(J) - ATAN(TANX)
          THED1(J)   = RTD*THE1(J)
          SNTHE1(J) = SIN(THE1(J))
          CSTHE1(J) = COS(THE1(J))
          CVF1(J)    = 1.0-EMO*CSTHE1(J)
 250  CONTINUE
C
C *** ROTOR 2
C
        DO 260 J = 1,NTH
          SINX = EMO*SIN(DTR*THAD2(J))
          COSX = SQRT(1.0-SINX*SINX)
          TANX = SINX/COSX
          THE2(J)    = DTR*THAD2(J) - ATAN(TANX)
          THED2(J)   = RTD*THE2(J)
          SNTHE2(J) = SIN(THE2(J))
          CSTHE2(J) = COS(THE2(J))
          CVF2(J)    = 1.0-EMO*CSTHE2(J)
 260  CONTINUE
C
C *** CALCULATE ROTOR - ROTOR GUST LOADING HARMONICS
C
        IF(IRW .LE. 0) GO TO 340
```

```
C
      DO 320 IZ = 1,NZUSE
        IZR  = IZ
        RMC1 = SQRT(YMC1(IZ)*YMC1(IZ)+ZMC1(IZ)*ZMC1(IZ))
        IF(RMC1 .LE. .0) GO TO 270
          PSI1  = ATAN2(ZMC1(IZ),YMC1(IZ))
          YMC1P = RMC1*COS(PSI1+DELBP1*DTR)
          ZMC1P = RMC1*SIN(PSI1+DELBP1*DTR)
          GO TO 280
 270  CONTINUE
          YMC1P = 0.
          ZMC1P = 0.
 280  CONTINUE
C
        RMC2 =SQRT(YMC2(IZ)*YMC2(IZ)+ZMC2(IZ)*ZMC2(IZ))
C
        IF(RMC2 .LE. .0) GO TO 290
          PSI2  = ATAN2(ZMC2(IZ),YMC2(IZ))
          YMC2P = RMC2*COS(PSI2+DELBP2*DTR)
          ZMC2P = RMC2*SIN(PSI2+DELBP2*DTR)
          GO TO 300
 290  CONTINUE
          YMC2P = 0.
          ZMC2P = 0.
 300  CONTINUE
C
        SXOCH(IZ) = ((ABS(XPCA)-ZMC1P-CHORD1(IZ)*
     &  SIN(BETPP1(IZ)*DTR)/2.)+
     &  DT2/DT1*(ZMC2P-CHORD2(IZ)*
     &  SIN(BETPP2(IZ)*DTR)/2.))/CHORD1(IZ)
C
        CALL RWVM(IZR,NSTEP,WTIV,THMOD1)
C
        CLCR = Z1(IZ)*EMT1/EMRR(IZ)
        OMCR = B1*CHORD2(IZ)*(EMT1*DT2/DT1+EMT2)/EMRR(IZ)
        IF(IPRNTW .NE. 0) WRITE(16,2120)IZ,EMRR(IZ),CLCR,OMCR
        IQMAXR = NWHM
C
        DO 310 IQ = 1,IQMAXR
          Q = IQ
          OMEGAR = Q*OMCR
C
          CALL SKCAL(OMEGAR,EMRR(IZ),SEARSR)
C
          IF(IPRNTW .NE. 0) WRITE(16,2140) OMEGAR,SEARSR
          RCLQ(IZ,IQ)  = TPI*CLCR*FCA(IQ)*SEARSR
          RPHIG(IZ,IQ) = OMEGAR*(1.-0.5*PI*PI/(1.+TPI*OMEGAR))
          NP12 = RPHIG(IZ,IQ)/TPI
          RPHIG(IZ,IQ) = RPHIG(IZ,IQ)-TPI*FLOAT(NP12)
 310    CONTINUE
C
        IF(IPRNTW.NE.0) WRITE(16,2180)(RCLQ(IZ,IQ),IQ=1,IQMAXR)
        IF(IPRNTW.NE.0) WRITE(16,2200)(RPHIG(IZ,IQ),IQ=1,IQMAXR)
 320  CONTINUE
C
      IF(IPRNTW.NE.0) WRITE(16,2240) (Z2(IZ),IZ=IZCL,NZUSE,IPRT)
C
      DO 330 IQ = 1,IQMAXR
```

```
          IF(IPRNTW.NE.0) WRITE(16,2260) (RCLQ(IZ,IQ),IZ=IZCL,
     &                  NZUSE,IPRT)
 330    CONTINUE
C
C ********************************************************************
C *** BEGIN  NOISE  CALCULATION ***************************************
C ********************************************************************
C
 340  CONTINUE
C
C *** INDEX OVER VISUAL OBSERVER ANGLE  -  J
C
        DO 600 J = 1,NTH
          ROD01 = YOD1/SNTHE1(J)
          IF(NSL.LT.1) ROD01 = ROD01*SIN(DTR*THAD1(J))
          ROD02 = YOD2/SNTHE2(J)
          IF(NSL.LT.1) ROD02 = ROD02*SIN(DTR*THAD2(J))
C
C *** INDEX OVER SPANWISE LOCATION INDEX  -  IZ
C
          DO 530 IZ = 1,NZUSE
            COSBP1 = COS(DTR*BETPP1(IZ))
            SINBP1 = SIN(DTR*BETPP1(IZ))
            COSBP2 = COS(DTR*BETPP2(IZ))
            SINBP2 = SIN(DTR*BETPP2(IZ))
            RMC1   = SQRT(YMC1(IZ)*YMC1(IZ)+ZMC1(IZ)*ZMC1(IZ))
            IF(RMC1 .LE. .00001) GO TO 350
                PSI1  = ATAN2(ZMC1(IZ),YMC1(IZ))
                YMC1P = RMC1*COS(PSI1+DELBP1*DTR)
                ZMC1P = RMC1*SIN(PSI1+DELBP1*DTR)
                GO TO 360
 350        CONTINUE
                YMC1P = 0.
                ZMC1P = 0.
 360        CONTINUE
            WU1    = Z1(IZ)*UT1*(1.-THETA1(IZ)*THMOD1)
            TB1    = WU1/VO
            BR1    = ATAN2(WU1,VO)
            THMC1 = 2.0*YMC1P/Z1(IZ)
            THTE1 = THMC1+CHORD1(IZ)*COSBP1/Z1(IZ)
            RMC2   = SQRT(YMC2(IZ)*YMC2(IZ)+ZMC2(IZ)*ZMC2(IZ))
            IF(RMC2 .LE. .00001) GO TO 370
                PSI2  = ATAN2(ZMC2(IZ),YMC2(IZ))
                YMC2P = RMC2*COS(PSI2+DELBP2*DTR)
                ZMC2P = RMC2*SIN(PSI2+DELBP2*DTR)
                GO TO 380
 370        CONTINUE
                YMC2P = 0.
                ZMC2P = 0.
 380        CONTINUE
            YMC2P  = RMC2*COS(PSI2+DELBP2*DTR)
            ZMC2P  = RMC2*SIN(PSI2+DELBP2*DTR)
            THMC2  = 2.0*YMC2P/Z2(IZ)
            THLE2  = THMC2-CHORD2(IZ)*COSBP2/Z2(IZ)
            THQC2  = THMC2-.5*CHORD2(IZ)*COSBP2/Z2(IZ)
            THMC2  = -THMC2
            THLE2  = -THLE2
            THQC2  = -THQC2
```

```
            CH1AX  = CHORD1(IZ)*SINBP1
            XQCHQC = ((ABS(XPCA)-ZMC1P-CHORD1(IZ)*SINBP1/2.)+
     &                DT2/DT1*(ZMC2P-CHORD2(IZ)*SINBP2/4.))/CH1AX
            DTHW1  = 2.*XQCHQC*TB1*CH1AX/Z1(IZ)
            THW1   = THTE1+DTHW1
            DELPHW = THW1-THQC2
            IF(IZ .EQ. 1) DPHTIP = DELPHW
            DELPHW = DELPHW-DPHTIP
C
C *** INDEX OVER HARMONIC NUMBER  -  M
C
            M1I = 1
            M1F = NHM1
C
            DO 420 M = M1I,M1F
                EM = M
                DPTH1R = 0.
                DPTH1I = 0.
                DPSL1R = 0.
                DPSL1I = 0.
                ROD1   = ROD01
                IF(INF .LT. 1) GO TO 400
                  IF(INFS .NE. 1) GO TO 390
                    ROD1 = ROD01-0.5*(HTR1+(1.-HTR1)*TPI*AKINF/(EM*B1*EMT1))
                    GO TO 400
 390              CONTINUE
                    ROD1 = ROD01-0.5*TPI*AKINF/(EM*B1*EMT1)
 400            CONTINUE
                PAMP1 = PCON1/(ROD1*CVF1(J))
                XF1   = EM*B1*EMT1/CVF1(J)
                PHIC  = 2.0*XF1*ROD1 - 0.5*EM*B1*PI
                NPH   = PHIC/TPI
                PHIC1 = PHIC - TPI*FLOAT(NPH)
                AKCON1 = 2.0*EM*B1/CVF1(J)
                FREQ(J,M) = EM*B1*RPS1
                IF(IDS .GT. 0) FREQ(J,M) = FREQ(J,M)/CVF1(J)
                SPLT(J,M) = 0.0
C
C *** WAVE NUMBER AND PHASE ANGLE CALCULATIONS
C
                AKX  = AKCON1*CHORD1(IZ)*EMT1/EMR1(IZ)
                AKY  = (AKX/EMT1)*(EMR1(IZ)*EMR1(IZ)*CSTHE1(J)-EMO)/Z1(IZ)
                PHIS = AKX*MCA1(IZ)/CHORD1(IZ)
                PHIO = AKY* FA1(IZ)/CHORD1(IZ)
                PHIT = PHIS+PHIO+PHIC
                NPH  = PHIT/TPI
                PHIT1  = PHIT - TPI*FLOAT(NPH)
                CSPHI1 = COS(PHIT1)
                SPHI1  = SIN(PHIT1)
C
C *** THICKNESS NOISE - ROTOR 1
C
                N = M*NB1
                X = XF1*Z1(IZ)*SNTHE1(J)
C
                CALL BJFN(X,N,BJNX)
C
                IF(BJNX .EQ. 0.0) GO TO 410
```

```
                    CALL AKVCAL(AKX,ITDO1,ALE1(IZ),BTE1(IZ),
     &     XTM1,DXTM1,AKVR,AKVI)
C
                    AMP    = ((EMR1(IZ)*AKX)**2)*TMOC1(IZ)*BJNX
                    DPTH1R = -AMP*(CSPHI1*AKVR-SPHI1*AKVI)
                    DPTH1I = -AMP*(SPHI1*AKVR+CSPHI1*AKVI)
C
C ***  STEADY LOADING NOISE - ROTOR 1
C
                    CALL AKLCAL(AKX,ILDO1,XLM1,AKLR,AKLI)
C
                    COEFU = (EM*B1)/X
                    FACTU = XF1*CHORD1(IZ)
                    AKYU  = 2.*FACTU*(COSBP1*CSTHE1(J)-
     &                      COEFU*SINBP1*SNTHE1(J))
                    IF(IHEL.EQ.1) AKYU = AKY
                    AMP    = (EMR1(IZ)**2)*(0.5*CL1(IZ)*AKYU)*BJNX
                    DPSL1R = -AMP*(SPHI1*AKLR+CSPHI1*AKLI)
                    DPSL1I = AMP*(CSPHI1*AKLR-SPHI1*AKLI)
 410        CONTINUE
                  DPMBR(IZ,M) = (DPTH1R + DPSL1R)*PAMP1
                  DPMBI(IZ,M) = (DPTH1I + DPSL1I)*PAMP1
C
 420      CONTINUE
C
C *** END OF LOOP FOR HARMONIC NUMBER INDEX - M
C
C
C *** STEADY LOADING AND THICKNESS NOISE - ROTOR 2
C
            M2I = NHM1+1
            M2F = NHM1+NHM2
C
            IF (IZ .GE. IZCL) THEN
            DO 450 M = M2I,M2F
                DPTH2R = 0.
                DPTH2I = 0.
                DPSL2R = 0.
                DPSL2I = 0.
                EM    = M-NHM1
                XF2   = EM*B2*EMT2/CVF2(J)
                ROD2 = ROD02
                IF(INF .LT. 1) GO TO 440
                  IF(INFS .NE. 1) GO TO 430
                    ROD2 = ROD02-0.5*(HTR2+(1.-HTR2)*TPI*AKINF/(EM*B2*EMT2))
                    GO TO 440
 430          CONTINUE
                    ROD2 = ROD02-0.5*TPI*AKINF/(EM*B2*EMT2)
 440        CONTINUE
                PAMP2  = PCON2/(ROD2*CVF2(J))
                PHIC   = 2.0*XF2*ROD2 - 0.5*EM*B2*PI
                NPH    = PHIC/TPI
                PHIC   = PHIC - TPI*FLOAT(NPH)
                AKCON2 = 2.0*EM*B2/CVF2(J)
                FREQ(J,M) = EM*B2*RPS2
                IF(IDS.GT.0) FREQ(J,M) = FREQ(J,M)/CVF2(J)
                SPLT(J,M) = 0.0
C
```

```
C *** WAVE NUMBER AND PHASE ANGLE CALCULATIONS
C
                AKX    = AKCON2*CHORD2(IZ)*(EMT2)/EMRR(IZ)
                AKY    = (AKX/EMT2)*(EMRR(IZ)*EMRR(IZ)*CSTHE2(J)-EMO)/Z2(IZ)
                PHIS   = AKX*MCA2(IZ)/CHORD2(IZ)
                PHIO   = AKY* FA2(IZ)/CHORD2(IZ)
                PHIT   = PHIS+PHIO+PHIC
                NPH    = PHIT/TPI
                PHIT2 = PHIT - TPI*FLOAT(NPH)
                CSPHI2 = COS(PHIT2)
                SPHI2  = SIN(PHIT2)
C
C *** THICKNESS NOISE - ROTOR 2
C
                N = (M-NHM1)*NB2
                X = XF2*Z2(IZ)*SNTHE2(J)
C
                CALL BJFN(X,N,BJNX)
C
                IF(BJNX .EQ. 0.0) GO TO 450
C
                  CALL AKVCAL(AKX,ITDO2,ALE2(IZ),BTE2(IZ),XTM2,
     &     DXTM2,AKVR,AKVI)
C
                  AMP = ((EMRR(IZ)*AKX)**2)*TMOC2(IZ)*BJNX
                  DPTH2R = -AMP*(CSPHI2*AKVR-SPHI2*AKVI)
                  DPTH2I = -AMP*(SPHI2*AKVR+CSPHI2*AKVI)
C
C *** STEADY LOADING NOISE - ROTOR 2
C
                  IDUM = IZ
C
                  CALL AKLCAL(AKX,ILDO2,XLM2,AKLR,AKLI)
C
                  COEFU = EM*B2/X
                  FACTU = XF2*CHORD2(IZ)
                  AKYU = 2.*FACTU*(COSBP2*CSTHE2(J)-COEFU*SINBP2*SNTHE2(J))
                  IF(IHEL .EQ. 1) AKYU = AKY
                  AMP = (EMRR(IZ)**2)*(0.5*CL2(IZ)*AKYU)*BJNX
                  DPSL2R = -AMP*(SPHI2*AKLR+CSPHI2*AKLI)
                  DPSL2I =  AMP*(CSPHI2*AKLR-SPHI2*AKLI)
                  DPMBR(IZ,M) = (DPTH2R + DPSL2R)*PAMP2
                  DPMBI(IZ,M) = (DPTH2I + DPSL2I)*PAMP2
 450    CONTINUE
C
C *** ROTOR - ROTOR UNSTEADY LOADING NOISE
C
            IF(IRW .LE. 0) GO TO 530
C
            DO 520 IQR = 1,IQMAXR
                QR = IQR
                PHI12L = QR*B1*DELPHW
                DO 510 IRR = 1,NHM2
                  DRULR = 0.
                  DRULI = 0.
                  RR    = IRR
                  M12   = NHM1+NHM2*IQR+IRR
                  N12   = IRR*NB2-IQR*NB1
```

```
                 EN12  = N12
                 FREQ(J,M12) = QR*B1*RPS1+RR*B2*RPS2
                 AKLQRT = 2.*AKINF*CO/(DT2*FREQ(J,M12))
                 ROD2  = ROD02
                 IF(INF .LT. 1) GO TO 470
                   IF(INFS .NE. 1) GO TO 460
                       ROD2 = ROD02-.5*(HTR2+AKLQRT*(1.-HTR2))
                       GO TO 470
 460             CONTINUE
                       ROD2 = ROD02-.5*AKLQRT
 470         CONTINUE
                 IF(IDS .GT. 0) FREQ(J,M12) = FREQ(J,M12)/CVF2(J)
                 SPLT(J,M12) = 0.0
C
                 CON12 = QR*B1*EMT1*DT2/DT1+RR*B2*EMT2
                 XRR   = CON12*Z2(IZ)*SNTHE2(J)/CVF2(J)
                 XRRA  = ABS(XRR)
                 N12A  = ABS(N12)
C
                 CALL BJFN(XRRA,N12A,BJNX12)
C
                 IF(BJNX12 .EQ. 0.) GO TO 500
                   AKCN12 = .5*B2*CON12/(TPI*ROD2*DT2)
                   IF(IHEL .NE. 1) GO TO 480
                       SINB2L = EMO/EMRR(IZ)
                       COSB2L = SQRT(1.-SINB2L*SINB2L)
                       GO TO 490
 480             CONTINUE
                       SINB2L = SINBP2
                       COSB2L = COSBP2
 490             CONTINUE
                   AKYU    = COSB2L*CSTHE2(J)-EN12*SINB2L*SNTHE2(J)/XRR
                   PAMP12  = AKCN12*AKYU*RCLQ(IZ,IQR)*BJNX12/
     &       (CVF2(J)*CVF2(J))
                   PAMP12  = PAMP12*GAMMA*PO*DT2*CHORD2(IZ)
     &       *EMRR(IZ)*EMRR(IZ)
                   AKLR    = 1.
                   AKLI    = 0.
                   PHSE12  = 0.
C
C *** PHASE ANGLE CALCULATIONS
C
                   CON12A  = (RR*B2+QR*B1*RPM1/RPM2)/CVF2(J)
                   CON12B  = (RR*B2-QR*B1)/Z2(IZ)
                   XUCP    = -.25
                   PHI12S  = -(CON12B*COSB2L+CON12A*EMT2*
     &       SINB2L*CSTHE2(J))
                   PHI12S  = PHI12S*2.*(MCA2(IZ)+CHORD2(IZ)*XUCP)
                   NP12    = PHI12S/TPI
                   PHI12S  = PHI12S-TPI*FLOAT(NP12)
                   PHI12O  = 2.*FA2(IZ)*(CON12B*SINB2L-CON12A*EMT2*
     &       COSB2L*CSTHE2(J))
                   NP12    = PHI12O/TPI
                   PHI12O  = PHI12O-TPI*FLOAT(NP12)
C
                   PHOBS   = 0.
                   PHI12C  = .5*PI+2.*ROD2*CON12/CVF2(J)+EN12*
     &       (PHOBS-.5*PI)
```

```
                    NP12     = PHI12C/TPI
                    PHI12C   = PHI12C-TPI*FLOAT(NP12)
                    PHSE12   = PHI12C-(PHI12S+PHI12O)+PHI12L+RPHIG(IZ,IQR)
                    NP12     = PHSE12/TPI
                    PHSE12   = PHSE12-TPI*FLOAT(NP12)
                    COSP12   = COS(PHSE12)
                    SINP12   = SIN(PHSE12)
                    DRULR    = PAMP12*COSP12
                    DRULI    = PAMP12*SINP12
 500          CONTINUE
                   DPMBR(IZ,M12) = DRULR
                   DPMBI(IZ,M12) = DRULI
 510        CONTINUE
C
 520      CONTINUE
C
C *** END OF LOOP FOR SPANWISE LOCATION INDEX - IZ
C
          END IF
 530      CONTINUE
C
C *** RADIAL INTEGRATION - ROTOR 1
C
            DO 550 M = M1I,M1F
              XI(1) = Z1(1)
              XI(2) = Z1(NZUSE)
              YI(1) = 0.0
C
              CALL LSPFIT(Z1,DPMBR(1,M),NZUSE,XI,YI,2,-1)
C
              PMBT1R = YI(2)
C
              CALL LSPFIT(Z1,DPMBI(1,M),NZUSE,XI,YI,2,-1)
C
              PMBT1I = YI(2)
              PMBTSQ = PMBT1R*PMBT1R+PMBT1I*PMBT1I
              IF(PMBTSQ .LE. 1.75E-13) GO TO 540
                  SPLT(J,M) = 127.58 + 10.0*ALOG10(PMBTSQ)
 540      CONTINUE
            IF(M.LE.5) BPHM(M,J,1) = SPLT(J,M)
              NHM = M
 550      CONTINUE
C
C *** RADIAL INTEGRATION - ROTOR 2 (SL&T)
C
            DO 570 M = M2I,M2F
              XI(1) = Z2(IZCL)
              XI(2) = Z2(NZUSE)
              YI(1) = 0.0
C
              CALL LSPFIT(Z2(IZCL),DPMBR(IZCL,M),NZCL,XI,YI,2,-1)
C
              PMBT2R = YI(2)
C
              CALL LSPFIT(Z2(IZCL),DPMBI(IZCL,M),NZCL,XI,YI,2,-1)
C
              PMBT2I = YI(2)
              PMBTSQ = PMBT2R*PMBT2R+PMBT2I*PMBT2I
```

```
            IF(PMBTSQ .LE. 1.75E-13) GO TO 560
                SPLT(J,M) = 127.58 + 10.0*ALOG10(PMBTSQ)
 560    CONTINUE
          IF(M-M2I+1.LE.5) BPHM(M-M2I+1,J,2) = SPLT(J,M)
            NHM = M
 570    CONTINUE
C
C *** RADIAL INTEGRATION - ROTOR 2 (UNSTEADY)
C
          IF(IRW .LE. 0) GO TO 600
C
          M12I = NHM1+NHM2+1
          M12F = NHM1+NHM2*(IQMAXR+1)
C
          DO 590 M = M12I,M12F
            XI(1) = Z2(IZCL)
            XI(2) = Z2(NZUSE)
            YI(1) = 0.0
C
            CALL LSPFIT(Z2(IZCL),DPMBR(IZCL,M),NZCL,XI,YI,2,-1)
C
            PMBRRR = YI(2)
C
            CALL LSPFIT(Z2(IZCL),DPMBI(IZCL,M),NZCL,XI,YI,2,-1)
C
            PMBRRI  = YI(2)
            PMBRRS = PMBRRR*PMBRRR+PMBRRI*PMBRRI
            IF(PMBRRS .LE. 1.75E-13) GO TO 580
                SPLT(J,M) = 127.58 + 10.0*ALOG10(PMBRRS)
 580    CONTINUE
            NHM = M
 590    CONTINUE
C
 600  CONTINUE
C
C *** END OF LOOP FOR VISUAL OBSERVER ANGLE INDEX - J
C
C--------------------------------------------------------------------
C
C     ---** PRINT OUT HARMONIC SPECTRA ---**
C
        WRITE(16,2620)
        IF(NSL.GT.0) WRITE(16,2640) DIST
        IF(NSL.LE.0) WRITE(16,2660) DIST
C
        JMAX=NTH
        JMAX1=9
        IF(NTH.LT.9) JMAX1=NTH
        WRITE(16,2680)(THAD1(J),J=1,JMAX)
        WRITE(16,2700)(THED1(J),J=1,JMAX)
        WRITE(16,2720)(THAD2(J),J=1,JMAX)
        WRITE(16,2740)(THED2(J),J=1,JMAX)
        WRITE(16,2760)
C
        DO 610 M=1,NHM
          WRITE(16,2820) M,FREQ(JMAX1,M),(SPLT(J,M),J=1,JMAX)
 610  CONTINUE
C
```

```
      WRITE(16,2800)
C
C *** CALCULATE 1/3 OCTAVE BAND LEVELS FROM SUMMED TONE HARMONICS
C
      CALL TOBSPL(SPLT,FREQ,NTH,NHM,SPL,IFREQ)
      IF(INF.EQ.0) CALL ATMOSP(SPL,NSL,DIST,THED1,NTH)
C
C *** PRINT OUT 1/3-OCTAVE SPECTRA
C
      IF(IPTO.LE.0) GO TO 620
        WRITE(16,2840)
        IF(NSL.GT.0) WRITE(16,2640) DIST
        IF(NSL.LE.0) WRITE(16,2660) DIST
        CALL SPLOUT(SPL,IFREQ,THAD,NTH)
 620  CONTINUE
C
      WRITE(16,2880)
C
 700  IF(INST.EQ.1) CALL CRPIE(ISOLAT)
C
C *** FORMAT STATEMENTS ------------------------------------------------
C
 1500   FORMAT
     & (//4X,'COUNTER-ROTATING PROPELLER NOISE PREDICTION PROGRAM')
 1540   FORMAT(/4X,'BETAP SURFACE APPROXIMATION MODEL',
     &   /4X,'WAKE MODEL INCLUDED')
 1560   FORMAT(/4X,'HELICOIDAL SURFACE APPROXIMATION MODEL',
     &   /4X,'WAKE MODEL INCLUDED')
 1600 FORMAT(/7X,'*** ROTOR 1 INPUT PARAMETERS ***')
 1620 FORMAT(/7X,'*** ROTOR 2 INPUT PARAMETERS ***')
 1640   FORMAT(/10X,28HPROPELLER TIP DIAMETER DT = F8.4,5H  FT.
     &          /10X,28HNUMBER OF BLADES NB        = I5
     &          /10X,28HSHAFT HORSEPOWER SHP       = F10.2
     &          /10X,28HROTATIVE SPEED (RPM)       = F10.1
     &          /10X,28HFLIGHT VELOCITY VO (FPS)   = F8.2
     &          /10X,28HAMBIENT TEMPERATURE TAMB   = F8.2,8H  DEG. F
     &          /10X,28HAMBIENT PRESSURE PAMB      = F8.3,5H PSIA
     &          /10X,28HNUMBER OF PROPELLERS NP    = I3)
 1660 FORMAT(//10X,'XPCA =',F7.4/10X,'IRW  =',I2,
     & /10X,'IWAKE=',I2,/10X,'IDS  =',I2,/10X,'INF  =',I2,
     & /10X,'INFS =',I2/)
 1680 FORMAT(/10X,20HTIP SPEED MACH NO. =F8.4,
     & /10X,20HFLIGHT    MACH NO. =F8.4,
     & /10X,20HHELICAL   MACH NO. =F8.4/)
 1840 FORMAT
     & (1H1//15X,'TABLE OF BLADE SECTION PROPERTIES - ROTOR 1',
     & //2X,7HSECTION,2X,4HR/RT,4X,5HCHORD,3X,4HTM/C,
     & 4X,5HBETAP,3X,3HMCA,5X,2HFA,6X,2HCL,6X,'CD'/)
 1880 FORMAT(I6,2X,3F8.4,F8.2,4F8.4)
 1980 FORMAT(//15X,'TABLE OF BLADE SECTION PROPERTIES - ROTOR 2',
     & //2X,7HSECTION,2X,4HR/RT,4X,5HCHORD,3X,4HTM/C,
     & 4X,5HBETAP,3X,3HMCA,5X,2HFA,6X,2HCL,/)
 2020 FORMAT(//15X,'ROTOR 1')
 2040 FORMAT(/15X,21HEFFECTIVE AERO SHP   = F8.2/
     & 15X,21HROTOR ANNULUS AREA  = F8.3,6H SQ FT/
     & 15X,21HDISK LOADING SHP/D2 = F8.2/
     & 15X,21HDISK LOADING SHP/AA = F8.2/
     & 15X,21HPOWER COEFFICIENT CP= F8.3/
```

```
     & 15X,21HBLADE TIP SPEED UT   = F8.2,' FT/SEC'/
     & 15X,21HBLADE PITCH ANGLE    = F8.2,' DEG.')
 2060 FORMAT(//15X,'ROTOR 2')
 2120 FORMAT(2X,'IZ=',I3,'  EMRR=',F10.4,'  CLCR=',F10.4,
     &    '  OMCR=',F10.4)
 2140 FORMAT(2X,'OMEGAR=',F10.4,'  SEARSR=',F10.4)
 2180 FORMAT(2X,'RCLQ=',10F10.4)
 2200 FORMAT(2X,'RPHIG=',10F10.4)
 2240 FORMAT(//15X,'ROTOR1 - ROTOR2 UNSTEADY LOADING LIFT
     & COEFFICIENTS',//30X,4HR/RT,//11F7.4//)
 2260 FORMAT(11E12.3)
 2620 FORMAT(1H1/4X,'*** TONE HARMONIC SPL SPECTRA ***'//)
 2640   FORMAT(15X,F10.2,2X,12HFT. SIDELINE//)
 2660   FORMAT(15X,F10.2,2X,7HFT. ARC     //)
 2680 FORMAT(2X,'THAO-R1    ',17F6.1)
 2700 FORMAT(2X,'THAE-R1    ',17F6.1)
 2720 FORMAT(2X,'THAO-R2    ',17F6.1)
 2740 FORMAT(2X,'THAE-R2    ',17F6.1)
 2760 FORMAT(12H    M    FREQ.)
 2800 FORMAT(///)
 2820 FORMAT(I4,F8.1,17F6.1)
 2840 FORMAT(1H1/4X,'*** 1/3-OCTAVE SPL SPECTRA ***'//)
 2880 FORMAT(///,'---------------------------------------------'///)
C
C
C ***********************************************************************
C *** END OF MAIN PROGRAM ***********************************************
C ***********************************************************************
C
      END
C
C ****************************
      SUBROUTINE READIN
C ***********************************************************************
C
C *************************************
C *** COMMONS/DIMENSIONS FOR CRPFAN ***
C *************************************
C
      REAL MCA,MCA1,MCA2
C
      COMMON /COM12/  NZUSE,DTR,RTD,VO,CO,SXOCH(51)
C
      COMMON /NLCOM/
     & AKINF,ALE(10),BETA34,BETAP(10),BTE(10),CHORD(10),
     & DELBP,DIST,DT,DXTM,HTR,ICOSZ,IDS,IHEL,ILDO,INF,INFS,
     & INST,IPTO,IRW,ISECT,ISOLAT,ITDO,NALE,NB,NBTE,NCASE,NHM,
     & NP,NSL,NSTEP,NTH,NRSEC,NZ,PAMB,PCTCL,RPM,RSEC(51),SCD(10),
     & SHP,SIGR,TAMB,THETA(10),THAD(17),TMOC(10),WTIV,XLM,
     & XPCA,XTM,YMC(10),Z(10),ZMC(10),Z34
C
     COMMON /INCOM/
     & NZ11,B341,DT1,NHM1,RPM1,SHP1,HTR1,ITDO1,XTM1,DXTM1,
     & YOD1,XLM1,ALE1(51),ALE11(10),BP11(10),BTE1(51),BTE11(10),
     & CHD11(10),ETA1,RSEC1(51),THAD1(17),TMOC11(10),YMC11(10),
     & Z11(10),ZMC11(10),ILDO1,
     & NZ21,B342,DT2,NHM2,RPM2,SHP2,HTR2,ITDO2,XTM2,DXTM2,
     & YOD2,XLM2,ALE2(51),ALE21(10),BP21(10),BTE2(51),BTE21(10),
```

```
     & CHD21(10),ETA2,RSEC2(51),THAD2(17),TMOC21(10),YMC21(10),
     & Z21(10),ZMC21(10),ILDO2
C
      COMMON /RWAKE1/ ISHAPE,BETAW,NWHM,VREF,IWAKE
C
      COMMON /HMONIC/ ST(300),VPN(300),FCA(21),FCB(21),FCDB(21)
C
      COMMON /PRINTD/ IPRT,IPR,IPRNTW
C
      COMMON /BLADE1/ THETA1(51), SIGR1(51), DPINT1(51), CHORD1(51),
     & TMOC1(51), Z1(51), ZMC1(51), YMC1(51), BETAP1(51), MCA1(51),
     & FA1(51), BETPP1(51), DELBP1, SCD1(51), EMT1, NB1
C
      COMMON /BLADE2/ THETA2(51), SIGR2(51) ,DPINT2(51), CHORD2(51),
     & TMOC2(51), Z2(51), ZMC2(51), YMC2(51), BETAP2(51), MCA2(51),
     & FA2(51), BETPP2(51), DELBP2, SCD2(51), EMT2, NB2
C
        COMMON /VTEX1/ ITPVTX,IHBVTX,TAU,VDUM1(3),VB1
C
        COMMON /VTEX3/ SBN(2),VDUM2(1205),VHTR,VDUM3(600)
C
        COMMON /VTEX4/ VDUM4(54),CI,TVTI
C
C ***      CI=CIRCULATION INDEX FOR TIP VORTEX
C ***       TVTI=TIP VORTEX TRAJECTORY INDEX
C
      DIMENSION FFDI(17), ALPHA(10), SCXD(10), XI(2), YI(2)
C
C ***********************************
C *** COMMONS/DIMENSIONS FOR CRPIE ***
C ***********************************
C
      DIMENSION ALD(51),CD(51),CL(51)
C
      DIMENSION ALDA(51),CDA(51),CLA(51)
C
      DIMENSION X(3),PSPANV(51)
C
      DIMENSION FUFV1(5),TF1V(5),TF2V(5),
     &          QF1V(5),QF2V(5),
     &          BSWPFD(51),BSWPAD(51)
C
      DIMENSION RADV(51),RADVA(51),CHDV(51),CHDVA(51),BETPV1(51),
     &          BETPV2(51),SIGV(51),SIGVA(51),
     &          FRV(51),FRVA(51),FRVAN(51),YSIGV(51),
     &          YSIGVA(51),POWF(51),
     &          POWA(51),FRDIM(51),YBETP1(51),
     &          YBETP2(51),XV(2),TMOCV(51),TMOCV2(51)
C
      DIMENSION THRFRV(51),THRRRV(51)
C
      DIMENSION YMCV1(51),YMCV2(51),ZMCV1(51),ZMCV2(51),
     &          CHD1(51),CHD2(51)
C
      DIMENSION RADDES(51),FRDES(51),YSGVD(51),YSGVDA(51),
     &          YBPD(51),YBPDA(51),THSRP(2,51),
     &          SIGVT(51),THSRPT(51),TTRT(51),QTRT(51),TDUT(51),
     &          RLV(51),XRVV(2,51),XRV(51),
```

```
     &           QDUT(51),QDVT(51),THCKV(51),THTDIV(20,2),THKVIF(51),
     &           THKVIR(51)
C
      DIMENSION TTR(2,51),QTR(2,51),TDUTR(2,51),TDVTR(2,51),
     &           QDUTR(2,51),QDVTR(2,51),TDVT(51)
C
      DIMENSION RADP(51),XLEP(51),XTEP(51),CAMPD(51),FRADP(51),
     &           XLEPD(51),XTEPD(51),CAMPDD(51)
C
      REAL       LAM,LAMA,LAM0,LAM2,MFLT,MTC
C
      COMMON /A/ ALD,CD,CL,LAM,FR,SIG,THETDF,DEGR,NPF,
     &            ALDA,CDA,CLA,LAMA,FRA,SIGA,NPA,UFR,
     &            UAFT,THETDA,NBF,NBA,SPR
C
      COMMON /B/ RRAT,FRDIM,YSIGV,YSIGVA,B701,B702,YBETP1,
     &            ASEPA,PI,LAM0,SPR0,PFRONT,PREAR,PCON1,PCON2,
     &            YBETP2,ASEP,N,ITMAX,ERRREL,RPROP
C
      COMMON /SRP/ NDES,RMINC,FRDES,SIGVT,THSRPT,RLV,XRV,THCKV,
     &              THFDG,ADVR,NBC,DTIP,CLACC,ARAC,AATC,WSS,
     &              ARAD,GAMC,PO,NHMAX,NMODES,FLEN,
     &              NPHLAG,NANG,THTDIV,ICALL,TTRT,QTRT,TDUT,
     &              NPGC,XWL,YWL,ZWL,MTC,FDIA,FRAWC,NTHTDI,
     &              TDVT,QDUT,QDVT,NROTC
C
      COMMON /PYLON/ APYLVW(51,51),PHPLVW(51,51),APYLCU(51,51),
     &                PHPLCU(51,51),APYLCV(51,51),PHPLCV(51,51),
     &                ARBDU
C
      COMMON /RCOM/  RHO,RPMF,RPMA,RTIPF,RTIPA,BET701,BET702,
     &                ASEPIN,MFLT,TO,XPCHA,XPCHAR,NPCLCD,
     &                NRAD,NRADA,NRADP,THETDP,CDP,IPYLON,
     &                RADV,CHDV,BETPV1,RADVA,CHDVA,BETPV2,
     &                RADP,XLEP,CAMPD,XTEP,PSPANV,NROT,
     &                TMOCV,TMOCV2,YMCV1,YMCV2,ZMCV1,ZMCV2,
     &                CHD1,CHD2
C
C ******************************************************
C ***        INPUT READS STANDARD CRUDFILES  -       ***
C *** IF INST = 1, INSTAL READS INSTALLATION FILE ***
C ******************************************************
C
      NAMELIST /INPUT/
     & AKINF,ALE,BETA34,BETAP,BETAW,BTE,CHORD,CI,DELBP,DIST,DT,DXTM,
     & HTR,ICOSZ,IDS,IHBVTX,IHEL,ILDO,INF,INFS,INST,IPRNTW,IPTO,IRW,
     & ISECT,ISHAPE,ISOLAT,ITDO,ITPVTX,IWAKE,NALE,NB,NBTE,NCASE,NHM,
     & NP,NSL,NSTEP,NTH,NRSEC,NWHM,NZ,PAMB,PCTCL,RPM,RSEC,SBN,SCD,SHP,
     & SIGR,TAMB,TAU,THETA,THAD,TMOC,TVTI,VO,VREF,WTIV,XLM,XPCA,XTM,
     & YMC,Z,ZMC,Z34
C
      NAMELIST /INSTAL/
     & NPCLCD,NDES,NRADP,THETDP,CDP,IPYLON,ALD,CL,CD,ALDA,CLA,CDA,
     & RADP,XLEP,CAMPD,XTEP,PSPANV,THFDG,CLACC,ARAC,AATC,WSS,NPGC,
     & XWL,YWL,ZWL,ARAD,NHMAX,NMODES,NROT,FLEN,FDIA,FRAWC,NPHLAG,
     & NANG,ARBDU
C
      PI  = 3.141592654
```

```
      DTR = PI/180.
      RTD = 1./DTR
C
C     --------------- INITIALIZE INPUT VARIABLES - ROTOR 1 ------------
C                            DEFAULT RUNNING PARAMETERS
C
        INST   = 0
        ISOLAT = 1
C
C *** READ INPUT/GEOMETRY FILE - ROTOR 1, FILE CODE = 10
C
      READ(10,INPUT,ERR=8000,END=9999)
C
C *** SET PROGRAM VARIABLES/CONSTANTS FOR ROTOR 1 FROM COMMON NAMELIST
C
      NZUSE = NZ
      DT1   = DT
      NB1   = NB
      VB1   = NB
      NHM1 = 10
      IF(NHM .NE. 0) NHM1 = NHM
      SHP1 = SHP
      RPM1 = RPM
      HTR1 = HTR
      ITDO1 = ITDO
      ILDO1 = ILDO
      XTM1  = XTM
      DXTM1 = DXTM
      YOD1 = DIST/DT1
      XLM1 = XLM
      NZ11 = NZ
      DELBP1 = DELBP
      IF(Z34 .EQ. 0) Z34 = .75
      Z341 = Z34
        DO 1180 ISL = 1,NZ
          Z1(ISL)       = Z(ISL)
          Z11(ISL)      = Z(ISL)
          ZMC1(ISL)     = ZMC(ISL)
          ZMC11(ISL)    = ZMC(ISL)
          YMC1(ISL)     = YMC(ISL)
          YMC11(ISL)    = YMC(ISL)
          CHORD1(ISL)   = CHORD(ISL)
          CHD11(ISL)    = CHORD(ISL)
          TMOC1(ISL)    = TMOC(ISL)
          TMOC11(ISL)   = TMOC(ISL)
          BETAP1(ISL)   = BETAP(ISL)
          BP11(ISL)     = BETAP(ISL)
          ALE1(ISL)     = ALE(ISL)
          ALE11(ISL)    = ALE(ISL)
          IF(NALE .NE. 1 .AND. NZ .LE. 1) GO TO 1170
            ALE1(ISL)  = ALE(1)
            ALE11(ISL) = ALE(1)
 1170     CONTINUE
          BTE1(ISL)     = BTE(ISL)
          BTE11(ISL)    = BTE(ISL)
          IF(NBTE .NE. 1 .AND. NZ .LE. 1) GO TO 1175
            BTE1(ISL)  = BTE(1)
            BTE11(ISL) = BTE(1)
```

```
 1175     CONTINUE
 1180   CONTINUE
 1190 CONTINUE
C
      DO 1200 ISL = 1,NZ11
        THETA1(ISL) = THETA(ISL)
        SCD1(ISL)   = SCD(ISL)
 1200 CONTINUE
C
      Z701   = HTR1 + 0.7*(1.-HTR1)
      XI(1)  = Z341
      XI(2)  = Z701
      CALL LSPFIT(Z11,BP11,NZ11, XI,YI,2, 0)
C
      BET341 = YI(1)
      TWIST1 = YI(1)-YI(2)
      IF(BETA34 .EQ. 0) GO TO 1210
        DELBP1  = BETA34-BET341
        B341 = BETA34
        GO TO 1215
 1210 CONTINUE
        B341 = BET341+DELBP1
 1215 CONTINUE
        BET701 = B341-TWIST1
C
C     -------------------------------------------------------
C     ----------------- INSERT SPANSECT HERE ------------------
C     -------------------------------------------------------
C
      IF(ISECT .EQ. 0) GO TO 1300
          IF(NRSEC .LE. 0) GO TO 1230
            DO 1220 JR=1,NRSEC
              RSEC1(JR)=RSEC(JR)
 1220       CONTINUE
            GO TO 1275
 1230     CONTINUE
            IF(ISECT.EQ.1) ISECT=51
            DRS=.5*PI/FLOAT(ISECT-1)
            XRS=0.
            DO 1240 JR=1,ISECT
              RSEC1(JR)=HTR1+(1.-HTR1)*COS(XRS)
              XRS=XRS+DRS
 1240       CONTINUE
            IF(ICOSZ .NE. 0) GO TO 1270
              DRS=(Z(1)-HTR1)/FLOAT(ISECT-1)
              RSEC1(1)=Z(1)
              DO 1260 JR=2,ISECT
                RSEC1(JR)=RSEC1(JR-1)-DRS
 1260         CONTINUE
 1270       CONTINUE
            NRSEC=ISECT
 1275     CONTINUE
C
          CALL LSPFIT(Z11,CHD11, NZ11,RSEC1,CHORD1,NRSEC,0)
          CALL LSPFIT(Z11,BP11,  NZ11,RSEC1,BETAP1,NRSEC,0)
          CALL LSPFIT(Z11,TMOC11,NZ11,RSEC1,TMOC1, NRSEC,0)
          CALL LSPFIT(Z11,ZMC11, NZ11,RSEC1,ZMC1,  NRSEC,0)
          CALL LSPFIT(Z11,YMC11, NZ11,RSEC1,YMC1,  NRSEC,0)
```

```
          CALL LSPFIT(Z11,ALE11, NZ11,RSEC1,ALE1,  NRSEC,0)
          CALL LSPFIT(Z11,BTE11, NZ11,RSEC1,BTE1,  NRSEC,0)
C
          DO 1280 IZ=1,NRSEC
            Z1(IZ)=RSEC1(IZ)
 1280     CONTINUE
C
          NZUSE  = NRSEC
 1290   CONTINUE
        CALL LSPFIT(Z11,SCD,  NZ11,RSEC1,SCD1,  NZUSE,0)
        CALL LSPFIT(Z11,THETA,NZ11,RSEC1,THETA1,NZUSE,0)
 1300 CONTINUE
      THMUL = 1.
C
      DO 1320 J = 1,NTH
        THAD1(J) = THAD(J)
        IF(XPCA .GE. 0) GO TO 1310
          TTH1 = TAN(THAD(J)*DTR)
          THAD1(J) = ATAN2(YOD1,(XPCA+YOD1/TTH1))*RTD
 1310 CONTINUE
        THTDIV(J,1) = THAD1(J)
 1320 CONTINUE
C
C ********************************************************************
C *** END OF ROTOR 1 INPUT DEFINITION *********************************
C ********************************************************************
C
C *** INITIALIZE INPUT VARIABLES - ROTOR 2
C
      NRSEC = 0
C
C *** READ INPUT/GEOMETRY FILE - ROTOR 2, FILE CODE = 11
C
      READ(11,INPUT,ERR=8020,END=9999)
C
C *** SET PROGRAM VARIABLES/CONSTANTS FOR ROTOR 2 FROM COMMON NAMELIST
C
       DT2 = DT
       NB2 = NB
       NHM2 = 10
       IF(NHM .NE. 0) NHM2 = NHM
       SHP2 = SHP
       RPM2 = RPM
       HTR2  = HTR
       VHTR  = HTR
       ITDO2 = ITDO
       ILDO2 = ILDO
       XTM2  = XTM
       DXTM2 = DXTM
       YOD2  = DIST/DT2
       XLM2  = XLM
       NZ21 = NZ
       DELBP2 = DELBP
       IF(Z34 .EQ. 0) Z34 = Z341*DT1/DT2
       Z342 = Z34
         DO 1340 ISL     = 1,NZ
           Z2(ISL)       = Z(ISL)
           Z21(ISL)      = Z(ISL)
```

```
            ZMC2(ISL)     = ZMC(ISL)
            ZMC21(ISL)    = ZMC(ISL)
            YMC2(ISL)     = YMC(ISL)
            YMC21(ISL)    = YMC(ISL)
            CHORD2(ISL)   = CHORD(ISL)
            CHD21(ISL)    = CHORD(ISL)
            TMOC2(ISL)    = TMOC(ISL)
            TMOC21(ISL)   = TMOC(ISL)
            BETAP2(ISL)   = BETAP(ISL)
            BP21(ISL)     = BETAP(ISL)
            ALE2(ISL)     = ALE(ISL)
            ALE21(ISL)    = ALE(ISL)
            IF(NALE .NE. 1) GO TO 1325
              IF(NZ .LE. 1) GO TO 1325
                ALE2(ISL)  = ALE(1)
                ALE21(ISL) = ALE(1)
 1325       CONTINUE
            BTE2(ISL)  = BTE(ISL)
            BTE21(ISL) = BTE(ISL)
            IF(NBTE .NE. 1) GO TO 1330
              IF(NZ .LE. 1) GO TO 1330
               BTE2(ISL)  = BTE(1)
               BTE21(ISL) = BTE(1)
 1330       CONTINUE
 1340     CONTINUE
 1350   CONTINUE
C
        DO 1360 ISL = 1,NZ21
         THETA2(ISL) = THETA(ISL)
 1360   CONTINUE
C
       Z702   = HTR2 + 0.7*(1.-HTR2)
       XI(1)  = Z342
       XI(2)  = Z702
C
       CALL LSPFIT(Z21,BP21,NZ21, XI,YI,2, 0)
C
       BET342 = YI(1)
       TWIST2 = YI(1)-YI(2)
        IF(BETA34 .EQ. 0) GO TO 1370
          DELBP2  = BETA34-BET342
          B342 = BETA34
          GO TO 1375
 1370   CONTINUE
          B342 = BET342+DELBP2
 1375   CONTINUE
         BET702 = B342-TWIST2
C      ---------------------------------------------------------
C      ----------------- INSERT SPANSECT HERE ------------------
C      ---------------------------------------------------------
        IF(ISECT .EQ. 0) GO TO 1460
            IF(NRSEC .LE. 0) GO TO 1385
              DO 1380 JR = 1,NRSEC
                RSEC2(JR) = RSEC(JR)
 1380         CONTINUE
              GO TO 1425
 1385       CONTINUE
              IF(ISECT .EQ. 1) ISECT = 51
```

```
            DRS = .5*PI/FLOAT(ISECT-1)
            XRS = 0.
            DO 1400 JR = 1,ISECT
              RSEC2(JR) = HTR2+(1.-HTR2)*COS(XRS)
              XRS = XRS+DRS
 1400       CONTINUE
            IF(ICOSZ .NE. 0) GO TO 1422
              DRS = (Z(1)-HTR2)/FLOAT(ISECT-1)
              RSEC2(1) = Z(1)
              DO 1420 JR = 2,ISECT
                RSEC2(JR) = RSEC2(JR-1)-DRS
 1420         CONTINUE
 1422       CONTINUE
            NRSEC = ISECT
 1425     CONTINUE
C
          CALL LSPFIT(Z21,CHD21, NZ21, RSEC2,CHORD2,NRSEC, 0)
          CALL LSPFIT(Z21,BP21,  NZ21, RSEC2,BETAP2,NRSEC, 0)
          CALL LSPFIT(Z21,TMOC21,NZ21, RSEC2,TMOC2, NRSEC, 0)
          CALL LSPFIT(Z21,ZMC21, NZ21, RSEC2,ZMC2,  NRSEC, 0)
          CALL LSPFIT(Z21,YMC21, NZ21, RSEC2,YMC2,  NRSEC, 0)
          CALL LSPFIT(Z21,ALE21, NZ21, RSEC2,ALE2,  NRSEC, 0)
          CALL LSPFIT(Z21,BTE21, NZ21, RSEC2,BTE2,  NRSEC, 0)
C
          DO 1440 IZ = 1,NRSEC
            Z2(IZ) = RSEC2(IZ)
 1440     CONTINUE
 1450   CONTINUE
      CALL LSPFIT(Z21,THETA,NZ21, RSEC2,THETA2,NZUSE, 0)
 1460 CONTINUE
      THMUL = 1.
C
      DO 1480 J = 1,NTH
        THAD2(J) = THAD(J)
        IF(XPCA .LE. 0) GO TO 1470
         TTH2 = TAN(THAD(J)*DTR)
         THAD2(J) = ATAN2(YOD1,XPCA+YOD1/TTH2)*RTD
 1470 CONTINUE
        THTDIV(J,2) = THAD2(J)
 1480 CONTINUE
C
C ***********************************************************************
C *** END OF ROTOR 2 INPUT DEFINITION ***********************************
C ***********************************************************************
C
        IF(INST.NE.1) RETURN
C
C ***********************************************************************
C *** REARRANGEMENT OF INPUT FOR 'CRPIE' USE ****************************
C ***********************************************************************
C
C
C *** SET STANDARD CONSTANTS
C
        RG        = 1716.2
        CALL GAMCAL(TAMB,GAMMA)
        GAMC      = GAMMA
        PO        = PAMB*144.
```

```
      TO        = TAMB+459.67
      CO        = SQRT(GAMMA*RG*TO)
      RHO       = PO/(RG*TO)
      RPMF      = RPM1
      RPMA      = RPM2
      RTIPF     = 6.*DT1
      RTIPA     = 6.*DT2
      NBF       = NB1
      PFRONT    = SHP1
      PREAR     = SHP2
      ASEPIN    = ABS(XPCA*DT1*12.)
      XPCHA     = 0.
      XPCHAR    = XPCA*DT1*12.
      IF(XPCA.LT.0.) THEN
          XPCHA  = XPCA*DT1*12.
          XPCHAR = 0.
      END IF
      MFLT      = VO/CO
      NBA       = NB2
      NRAD      = NZUSE
      NRADA     = NZUSE
      NTHTDI    = NTH
      DO 1500 I=1,NZUSE
        FRV(I)     = Z1(NZUSE+1-I)
        RADV(I)    = 6.*DT1*Z1(NZUSE+1-I)
        CHDV(I) = 12.*DT1*CHORD1(NZUSE+1-I)
        BETPV1(I) = BETAP1(NZUSE+1-I)
        YMCV1(I)   = YMC1(NZUSE+1-I)
        ZMCV1(I)   = ZMC1(NZUSE+1-I)
        TMOCV(I)   = TMOC1(NZUSE+1-I)
        CHD1(I)    = CHORD1(NZUSE+1-I)
 1500 CONTINUE
      DO 1510 I=1,NZUSE
        FRVA(I)     = Z2(NZUSE+1-I)
        RADVA(I)    = 6.*DT2*Z2(NZUSE+1-I)
        CHDVA(I)    = 12.*DT2*CHORD2(NZUSE+1-I)
        BETPV2(I)   = BETAP2(NZUSE+1-I)
        YMCV2(I)    = YMC2(NZUSE+1-I)
        ZMCV2(I)    = ZMC2(NZUSE+1-I)
        TMOCV2(I)   = TMOC2(NZUSE+1-I)
        CHD2(I)     = CHORD2(NZUSE+1-I)
 1510 CONTINUE
C
C *** READ INSTALLATION EFFECTS INPUT - FILE CODE 12
C
      READ(12,INSTAL,ERR=8040,END=9999)
C
      IF(ISOLAT.EQ.1) ARAD = DIST
C
C *** OUTPUT OF INSTALLATION EFFECTS INPUTS
C
C****Error checks for NRAD,NRADA,NRADP,NPCLCD,,NDES (all to be
C**** less than or equal to 51) and for NTHTDI (.LE. 20)
C
    IF(NRAD.LE.51) GO TO 1520
      WRITE(14,7000)
      GO TO 9999
 1520 CONTINUE
```

```
      IF(NRADA.LE.51) GO TO 1530
        WRITE(14,7010)
        GO TO 9999
 1530 CONTINUE
      IF(NRADP.LE.51) GO TO 1540
        WRITE(14,7020)
        GO TO 9999
 1540 CONTINUE
      IF(NPCLCD.LE.51) GO TO 1550
        WRITE(14,7030)
        GO TO 9999
 1550 CONTINUE
      IF(NDES.LE.51) GO TO 1560
        WRITE(14,7040)
        GO TO 9999
 1560 CONTINUE
      IF(NTHTDI.LE.20) GO TO 1570
        WRITE(14,7050)
        GO TO 9999
 1570 CONTINUE
C
C****Done with error checks
C****Reset values of NHMAX,NMODES and NPGC if too high
C
      IF(NHMAX.GT.5)NHMAX = 5
      IF(NMODES.GT.10)NMODES = 10
      IF(NPGC.GT.64)NPGC = 64
C
C *** WRITE OUT PROGRAM NAME HEADER
C
      WRITE(14,7060)
C
C *** WRITE OUT INPUT PARAMETERS
C
      WRITE(14,7070)
C
      WRITE(14,7080) RHO,TAMB,PAMB,GAMC,MFLT,ASEPIN,ARBDU,AATC
C
      WRITE(14,7090) THFDG
C
      IF(IPYLON.EQ.0) THEN
        WRITE(14,7140)
      ELSE
C
      WRITE(14,7100)
C
      WRITE(14,7110) CDP,THETDP
C
      WRITE(14,7120)
C
      DO 1580 J = 1,NRADP
        RADP(J) = RADP(J)*RTIPF
        XLEP(J) = XPCHA - XLEP(J)*DT1*12.
        XTEP(J) = XPCHA - XTEP(J)*DT1*12.
        WRITE(14,7130) RADP(J),XLEP(J),XTEP(J),CAMPD(J)
 1580 CONTINUE
      END IF
C
```

```
      WRITE(14,7150)
C
      WRITE(14,7160) RPMF,RTIPF,NBF,BET701,PFRONT,XPCHA
C
      IF(NROT.EQ. 1)WRITE(14,7170)
      IF(NROT.EQ.-1)WRITE(14,7180)
      WRITE(14,7190)
      WRITE(14,7200)
      DO 1590 J = 1,NPCLCD
        WRITE(14,7210) ALD(J),CL(J),CD(J)
 1590 CONTINUE
C
      WRITE(14,7220)
      WRITE(14,7230)
      DO 1600 J = 1,NRAD
        WRITE(14,7240) RADV(J),CHDV(J),BETPV1(J),YMCV1(J),ZMCV1(J),
     &                          TMOCV(J)
 1600 CONTINUE
C
      WRITE(14,7250)
      DO 1610 J = 1,NDES
        WRITE(14,7260) PSPANV(J)
 1610 CONTINUE
C
      WRITE(14,7270)
C
      WRITE(14,7160) RPMA,RTIPA,NBA,BET702,PREAR,XPCHAR
C
      NROTM=-NROT
      IF(NROTM.EQ. 1)WRITE(14,7170)
      IF(NROTM.EQ.-1)WRITE(14,7180)
      WRITE(14,7190)
      WRITE(14,7200)
      DO 1620 J = 1,NPCLCD
        WRITE(14,7210) ALDA(J),CLA(J),CDA(J)
 1620 CONTINUE
C
      WRITE(14,7220)
      WRITE(14,7230)
      DO 1630 J = 1,NRADA
        WRITE(14,7240) RADVA(J),CHDVA(J),BETPV2(J),YMCV2(J),ZMCV2(J),
     &                          TMOCV2(J)
 1630 CONTINUE
C
      WRITE(14,7280)CLACC,ARAC,WSS
      WRITE(14,7290)FLEN,FDIA,FRAWC
      WRITE(14,7300)XWL,YWL,ZWL
      WRITE(14,7310)
C
      RETURN
C
C --- FORMAT STATEMENTS ----------------------------------------------
C
 7000 FORMAT(1X,'INPUT ERROR - NRAD .GT. 50')
 7010 FORMAT(1X,'INPUT ERROR - NRADA .GT. 50')
 7020 FORMAT(1X,'INPUT ERROR - NRADP .GT. 50')
 7030 FORMAT(1X,'INPUT ERROR - NPCLCD .GT. 50')
 7040 FORMAT(1X,'INPUT ERROR - NDES .GT. 50')
```

```
 7050 FORMAT(1X,'INPUT ERROR - NTHTDI .GT. 20')
 7060 FORMAT(38X,'- C R P I E -',
    &//20X,'COUNTER ROTATION PROPELLER INSTALLATION EFFECTS PROGRAM')
 7070 FORMAT(//24X,'* * *   I N P U T    P A R A M E T E R S   * * *',
    &                      ///////)
 7080 FORMAT(/3X,'DENSITY (slug/cu.ft.)      =  ',F10.6,
     &       /3X,'TEMP.(STATIC)- deg. F      =  ',F6.2,
     &       /3X,'PRESSURE(STATIC) psia      =  ',F7.3,
     &       /3X,'GAMMA                      =  ',F7.3,
     &       /3X,'FLIGHT MACH #              =  ',F8.4,
     &       /3X,'AXIAL SEP. OF PCA (ins.)  =  ',F7.3,
     &       /3X,'ARBITRARY DELTA (u/U)      =  ',F7.3,
     &       /3X,'PROP. ANG. OF ATT. (deg.) =  ',F6.2)
 7090 FORMAT(//1X,'  REPRESENTATIVE THICKNESS / MAX THICKNESS :',F6.3,
     &        /1X,'   (USED FOR THICKNESS NOISE CALCULATIONS)')
 7100 FORMAT('1',//10X,'  PYLON RELATED INPUTS              ')
 7110 FORMAT(/3X,'PYLON VISCOUS CD          =  ',F8.4,
     &       /3X,'ANG. LOC. OF PYLON (deg.)=  ',F6.2)
 7120 FORMAT(/3X,'RAD(ins.)',5X,'XLE(ins.)',6X,'XTE(ins.)',6X,
     &'CMBR(deg.)',/)
 7130 FORMAT(3X,F7.3,8X,F7.2,8X,F7.2,8X,F7.3)
 7140 FORMAT(/10X,' PYLON EFFECT NOT INCLUDED IN RESULTS')
 7150 FORMAT('1',//10X,'  FRONT  ROTOR           ')
 7160 FORMAT(/3X,'RPM                        =  ',F8.1,
     &       /3X,'TIP RADIUS (ins.)          =  ',F8.3,
     &       /3X,'NUMBER OF BLADES           =  ',I8,
     &       /3X,'70 % SPAN PITCH (deg.)     =  ',F8.3,
     &       /3X,' (initial estimate)           ',
     &       /3X,'POWER ABSORBED (HP)        =  ',F8.1,
     &       /3X,'AXIAL LOC. OF PCA (ins.) =  ',F8.2)
 7170 FORMAT (3X,'PROP. ROTATION (FLA)       =   COUNTER-CLOCKWISE')
 7180 FORMAT (3X,'PROP. ROTATION (FLA)       =   CLOCKWISE')
 7190 FORMAT(/10X,'  CL and CD VERSUS ANGLE OF ATTACK (deg.)')
 7200 FORMAT(/1X,'    ANGLE',8X,'    CL',8X,'    CD',/)
 7210 FORMAT(3X,F7.5,8X,F7.5,8X,F7.5)
 7220 FORMAT(/10X,'  BLADE GEOMETRY                  ')
 7230 FORMAT(/3X,'RAD(ins.)',6X,'CHORD(ins.)',3X,'BETAP(deg.)',6X,
     &'YMC/DT',8X,'ZMC/DT',10X,'TMOC',/)
 7240 FORMAT(3X,F7.3,8X,F7.3,8X,F7.2,8X,F7.4,8X,F7.4,8X,F7.4)
 7250 FORMAT('1',//10X,' PER CENT SPAN LOCATIONS TO BE USED FOR',
     &                 ' ACOUSTIC CALCULATIONS',//)
 7260 FORMAT(33X,F6.2,' %')
 7270 FORMAT('1',//10X,'  REAR   ROTOR           ')
 7280 FORMAT('1',//1X,'AIRCRAFT WING PARAMETERS:',
     &          8X,'LIFT COEFFICIENT  =  ',F6.3,
     &        /34X,'ASPECT RATIO      =  ',F6.3,
     &        /34X,'SEMI-SPAN (ft.)   =  ',F6.3)
 7290 FORMAT(//1X,'AIRCRAFT FUSELAGE PARAMETERS:',
     &          4X,'TOTAL LENGTH (ft.)     =  ',F6.3,
     &        /34X,'MAX. DIAMETER (ft.)    =  ',F6.3,
     &        /34X,'FRAC. AHEAD OF WING LL =  ',F6.3)
 7300 FORMAT(//1X,'PROPELLER DISK CENTER LOCATION:',
     &          2X,'X DIST. DOWNSTREAM OF WING LL (ft.)      =  ',F6.3,
     &        /34X,'Y DIST. FROM A/C CL ALONG WING LL (ft.) =  ',F6.3,
     &        /34X,'Z DIST. ABOVE WING LL (ft.)              =  ',F6.3)
 7310 FORMAT(////20X,'*****END WRITE OF INPUT PARAMETERS*****')
C
C -----------------------------------------------------------------
```

```
C
C *** READ ERRORS ***
C
 8000 WRITE (15,8010)
 8010 FORMAT(1X,'ERROR READING NAMELIST INPUT - FILE CODE = 10')
      GO TO 9999
 8020 WRITE (15,8030)
 8030 FORMAT(1X,'ERROR READING NAMELIST INPUT - FILE CODE = 11')
      GO TO 9999
 8040 WRITE (15,8050)
 8050 FORMAT(1X,'ERROR READING NAMELIST INPUT - FILE CODE = 12')
C
 9999 STOP
      END
C ****************************************
      SUBROUTINE GAMCAL(TAMB,GAMMA)
C **********************************************************************
C
C     SPECIFIC HEAT RATIO CALCULATION FOR AIR AT LOW PRESSURES
C               FROM KEENAN AND KAYE GAS TABLES
C
C **********************************************************************
C
      TC=-25.0
      GAMMA=1.402
      IF(TAMB.LT.TC) GO TO 110
      IF(TAMB.GT.200.0) GO TO 100
        GAMMA = 1.402 - 0.000017778*(TAMB+25.0)
        GO TO 110
 100  CONTINUE
      GAMMA=1.398 - 0.000043333*(TAMB - 200.0)
 110  RETURN
      END
C
C **************************
      SUBROUTINE PITCH1
C **********************************************************************
C
C       CALCULATE BLADE COORDINATE CHANGES DUE TO
C       PITCH ANGLE CHANGE RELATIVE TO 'DESIGN' SETTING
C       --- ALSO CALCULATE FACE ALIGNMENT (FA) AND MID-CHORD
C           ALIGNMENT (MCA) ---
C
C **********************************************************************
C
      COMMON /COM12/ NZ,DTR,RTD,VO,CO,SXOCH(51)
C
      COMMON /BLADE1/ THETA1(51), SIGR1(51), DPINT1(51),
     & CHORD1(51), TMOC1(51), Z1(51), ZMC1(51), YMC1(51),
     & BETAP1(51), MCA1(51), FA1(51), BETPP1(51), DELBP1, SCD1(51)
C
      REAL MCA1
C
      PI = 3.1415926
      DBP1R = DTR*DELBP1
C
      DO 100 I = 1,NZ
C
```

```
      DZ  =  0.5*CHORD1(I)*SIN(DTR*BETAP1(I))
      DY  =  0.5*CHORD1(I)*COS(DTR*BETAP1(I))
      ZLE = ZMC1(I)-DZ
      ZTE = ZMC1(I)+DZ
      YLE = YMC1(I)-DY
      YTE = YMC1(I)+DY
C
      RLE = SQRT(ZLE*ZLE+YLE*YLE)
      RMC = SQRT(ZMC1(I)**2+YMC1(I)**2)
      RTE = SQRT(ZTE*ZTE+YTE*YTE)
C
      IF(ZMC1(I) .EQ. 0.0) GO TO 20
        PMC = ATAN2(YMC1(I),ZMC1(I))
        GO TO 30
 20   CONTINUE
      PMC = 0.0
      IF(YMC1(I) .LT. 0.0) PMC = -0.5*PI
      IF(YMC1(I) .GT. 0.0) PMC =  0.5*PI
 30   CONTINUE
      IF(ZLE .EQ. 0.0) GO TO 40
        PLE = ATAN2(YLE,ZLE)
        GO TO 50
 40   CONTINUE
      PLE = 0.0
      IF(YLE .LT. 0.0) PLE = -0.5*PI
      IF(YLE .GT. 0.0) PLE =  0.5*PI
 50   CONTINUE
      IF(ZTE .EQ. 0.0) GO TO 60
        PTE = ATAN2(YTE,ZTE)
        GO TO 70
 60   CONTINUE
      PTE = 0.0
      IF(YTE .LT. 0.0) PTE = -0.5*PI
      IF(YTE .GT. 0.0) PTE =  0.5*PI
 70   CONTINUE
      PPMC = PMC-DBP1R
      PPLE = PLE-DBP1R
      PPTE = PTE-DBP1R
C
      YMCP = RMC*SIN(PPMC)
      YLEP = RLE*SIN(PPLE)
      YTEP = RTE*SIN(PPTE)
C
      ZMCP = RMC*COS(PPMC)
      ZLEP = RLE*COS(PPLE)
      ZTEP = RTE*COS(PPTE)
C
      TOP  = ZTEP - ZLEP
      BOT  = YTEP-YLEP
      IF(BOT .EQ. 0.0) GO TO 80
        BETPP1(I) = ATAN2(TOP,BOT)
        GO TO 90
 80   CONTINUE
      BETPP1(I) = 0.0
      IF(TOP .LT. 0.0) BETPP1(I) = -0.5*PI
      IF(TOP .GT. 0.0) BETPP1(I) =  0.5*PI
 90   CONTINUE
C
```

```
        COSBPP   = COS(BETPP1(I))
        SINBPP   = SIN(BETPP1(I))
        FA1(I)   = ZMCP*COSBPP - YMCP*SINBPP
        MCA1(I) = ZMCP*SINBPP + YMCP*COSBPP
        BETPP1(I) = RTD*BETPP1(I)
C
 100   CONTINUE
C
      RETURN
      END
C
C ***************************
      SUBROUTINE PITCH2
C ***********************************************************************
C
C       CALCULATE BLADE COORDINATE CHANGES DUE TO
C       PITCH ANGLE CHANGE RELATIVE TO 'DESIGN' SETTING
C       --- ALSO CALCULATE FACE ALLIGNMENT (FA) AND MID-CHORD
C           ALLIGNMENT (MCA) ---
C
C ***********************************************************************
C
      COMMON /COM12/ NZ,DTR,RTD,VO,CO,SXOCH(51)
      COMMON /BLADE2/ THETA2(51), SIGR2(51), DPINT2(51),
     & CHORD2(51), TMOC2(51), Z2(51), ZMC2(51), YMC2(51),
     & BETAP2(51), MCA2(51), FA2(51), BETPP2(51), DELBP2, SCD2(51)
C
      REAL MCA2
C
      PI = 3.1415926
      DBP2R = DTR*DELBP2
C
C
      DO 100 I = 1,NZ
C
        DZ   = 0.5*CHORD2(I)*SIN(DTR*BETAP2(I))
        DY   = 0.5*CHORD2(I)*COS(DTR*BETAP2(I))
        ZLE = ZMC2(I)-DZ
        ZTE = ZMC2(I)+DZ
        YLE = YMC2(I)-DY
        YTE = YMC2(I)+DY
C
        RLE = SQRT(ZLE*ZLE+YLE*YLE)
        RMC = SQRT(ZMC2(I)**2+YMC2(I)**2)
        RTE = SQRT(ZTE*ZTE+YTE*YTE)
C
        IF(ZMC2(I) .EQ. 0.0) GO TO 20
          PMC = ATAN2(YMC2(I),ZMC2(I))
          GO TO 30
 20     CONTINUE
        PMC = 0.0
        IF(YMC2(I) .LT. 0.0) PMC = -0.5*PI
        IF(YMC2(I) .GT. 0.0) PMC =  0.5*PI
 30     CONTINUE
        IF(ZLE .EQ. 0.0) GO TO 40
          PLE = ATAN2(YLE,ZLE)
          GO TO 50
 40     CONTINUE
```

```
      PLE = 0.0
      IF(YLE .LT. 0.0) PLE = -0.5*PI
      IF(YLE .GT. 0.0) PLE =  0.5*PI
 50   CONTINUE
      IF(ZTE .EQ. 0.0) GO TO 60
        PTE = ATAN2(YTE,ZTE)
        GO TO 70
 60   CONTINUE
      PTE=0.0
      IF(YTE .LT. 0.0) PTE = -0.5*PI
      IF(YTE .GT. 0.0) PTE =  0.5*PI
 70   CONTINUE
      PPMC = PMC-DBP2R
      PPLE = PLE-DBP2R
      PPTE = PTE-DBP2R
C
      YMCP = RMC*SIN(PPMC)
      YLEP = RLE*SIN(PPLE)
      YTEP = RTE*SIN(PPTE)
C
      ZMCP = RMC*COS(PPMC)
      ZLEP = RLE*COS(PPLE)
      ZTEP = RTE*COS(PPTE)
C
      TOP = ZTEP - ZLEP
      BOT = YTEP-YLEP
      IF(BOT .EQ. 0.0) GO TO 80
        BETPP2(I) = ATAN2(TOP,BOT)
        GO TO 90
 80   CONTINUE
      BETPP2(I) = 0.0
      IF(TOP .LT. 0.0) BETPP2(I) = -0.5*PI
      IF(TOP .GT. 0.0) BETPP2(I) =  0.5*PI
 90   CONTINUE
C
      COSBPP = COS(BETPP2(I))
      SINBPP = SIN(BETPP2(I))
      FA2(I)  = ZMCP*COSBPP - YMCP*SINBPP
      MCA2(I) = ZMCP*SINBPP + YMCP*COSBPP
      BETPP2(I) = RTD*BETPP2(I)
C
 100  CONTINUE
C
      RETURN
      END
C
C *********************************************************
      SUBROUTINE TOBSPL(SPLT,FREQ,NTH,NHM,SPL,IFREQ)
C ***********************************************************************
C
C      1/3-OCTAVE LEVEL CALCULATION FROM TONE HARMONIC SPECTRUM
C
C ***********************************************************************
C
      DIMENSION SPLT(17,230),FREQ(17,230),SPL(17,27),IFREQ(27)
C
      DO 30 J = 1,NTH
        FL = 44.545
```

```
        DO 20 I = 1,27
          FC  = IFREQ(I)
          FH  = 1.122462*FC
          SUM = 0.0
          SPL(J,I) = 0.0
          DO 10 M = 1,NHM
            FT = FREQ(J,M)
            IF(FT .LT. FL) GO TO 10
              IF(FT .GE. FH) GO TO 10
                IF(SPLT(J,M) .LE. 0.0) GO TO 10
                  SUM = SUM+10.0**(SPLT(J,M)/10.0)
 10       CONTINUE
          FL = FH
          IF(SUM .LE. 0.0) GO TO 20
            SPL(J,I) = 10.0*ALOG10(SUM)
 20     CONTINUE
 30   CONTINUE
C
      RETURN
      END
C
C ************************************************
      SUBROUTINE SPLOUT(SPL,IFREQ,THAD,NTH)
C *****************************************************************
C
C     CALCULATE DBA, DBD, AND  PRINT OUT 1/3-OCTAVE SPL TABLES
C
C *****************************************************************
C
      DIMENSION SPL(17,27),IFREQ(27)
      DIMENSION THAD(17),OASPL(17),IANG(17)
      DIMENSION WDBA(27),WDBD(27),DBA(17),DBD(17)
C
      COMMON /COM12/ NZ,DTR,RTD,VO,CO,SXOCH(51)
C
C *** A - WEIGHTING FOR 50 Hz - 20 KHz 1/3 OCTAVE BANDS
C
      DATA WDBA/
     & -30.2,-26.2,-22.5,-19.1,-16.1,-13.4,-10.9,-8.6,-6.6,
     & -4.8,-3.2,-1.9,-0.8,0.0,0.6,1.0,1.2,1.3,1.2,1.0,0.5,
     & -0.1,-1.1,-2.5,-4.3,-6.6,-9.3/
C
C *** D - WEIGHTING FOR 50 Hz - 20 KHz 1/3 OCTAVE BANDS
C
      DATA WDBD/
     & -12.8,-10.9,-9.0,-7.2,-5.5,-4.0,-2.6,-1.6,-0.8,-0.4,
     & -0.3,-0.5,-0.6,0.0,2.0,4.9,7.9,10.4,11.6,11.1,9.6,7.6,
     & 5.5,3.4,1.4,-0.7,-2.7/
C
      DO 10 J = 1,NTH
       IANG(J)=THAD(J)
 10   CONTINUE
C
      J1 = 1
      J2 = NTH
C
C *** CALCULATE DBA AND DBD(APPROXIMATE PNL)
C
```

```
      DO 30 J = J1,J2
        SUMO = 0.0
        SUMA = 0.0
        SUMD = 0.0
        OASPL(J) = 0.0
        DBA(J)   = 0.0
        DBD(J)   = 0.0
        DO 20 I = 1,27
          IF(SPL(J,I) .LE. 0.0) GO TO 20
            SUMO = SUMO+10.0**(SPL(J,I)/10.0)
            SLA  = SPL(J,I) + WDBA(I)
            SLD  = SPL(J,I) + WDBD(I)
            SUMA = SUMA+10.0**(SLA/10.0)
            SUMD = SUMD+10.0**(SLD/10.0)
 20     CONTINUE
        IF(SUMO .GT. 0.0) OASPL(J) = 10.0*ALOG10(SUMO)
        IF(SUMA .GT. 0.0) DBA(J)   = 10.0*ALOG10(SUMA)
        IF(SUMD .GT. 0.0) DBD(J)   = 10.0*ALOG10(SUMD)
 30   CONTINUE
C
      WRITE(16,1000) (IANG(J),J = J1,J2)
      WRITE(16,1010)
C
      DO 40 I=1,27
        WRITE(16,1020) IFREQ(I),(SPL(J,I),J = J1,J2)
 40   CONTINUE
C
      WRITE(16,1030) (OASPL(J),J = J1,J2)
      WRITE(16,1040) (DBA(J),J = J1,J2)
      WRITE(16,1050) (DBD(J),J = J1,J2)
C
      WRITE(16,1060)
C
      RETURN
C
C *** FORMAT STATEMENTS ------------------------------------------------
C
 1000 FORMAT(1X,'O.ANG',1X,17I6)
 1010 FORMAT(2X,'FREQ')
 1020 FORMAT(1X,I5,1X,18F6.1)
 1030 FORMAT(/1X,'OASPL',1X,18F6.1)
 1040 FORMAT(1X,' DBA ',1X,18F6.1)
 1050 FORMAT(1X,' DBD ',1X,18F6.1)
 1060 FORMAT(///)
C
C ----------------------------------------------------------------------
C
       END
C
C ****************************************************************
      SUBROUTINE AKVCAL(AKX,ITDO,ALE,BTE,XTM,DXTM,AKVR,AKVI)
C **********************************************************************
C
C      THIS SUBROUTINE CALCULATES THE THICKNESS OR VOLUME
C      DISPLACEMENT NOISE SOURCE CHORDWISE NON-COMPACTNESS
C      FACTOR     --- K-SUB-V ---
C
C **********************************************************************
```

```
C
C *** POINT-IMPULSIVE DISTRIBUTION ( ITDO = 0 )
C
      AKVR = 1.0
      AKVI = 0.0
      IF(ITDO .GT. 0) GO TO 10
        GO TO 100
C
C *** RECTANGULAR DISTRIBUTION ( ITDO = 1 )
C
 10   CONTINUE
      IF(ITDO .GT. 1) GO TO 20
        IF(AKX .LE. 0.0) GO TO 100
          X    = ABS(0.5*AKX)
          AKVR = (SIN(X))/X
          GO TO 100
C
C *** PARABOLIC DISTRIBUTION ( ITDO = 2 )
C
 20   CONTINUE
      IF(ITDO .GT. 2) GO TO 30
        AKVR = 0.66667
        AKVI = 0.0
        IF(AKX .LE. 0.0) GO TO 100
          X    = ABS(0.5*AKX)
          AKVR = 2.0*((SIN(X))/X - COS(X))/(X**2)
          GO TO 100
C
C *** TRIANGULAR DISTRIBUTION ( ITDO = 3 )
C
 30   CONTINUE
      IF(ITDO .GT. 3) GO TO 40
        AKVR = 0.5
        AKVI = 0.0
        IF(AKX .LE. 0.0) GO TO 100
          X    = ABS(0.5*AKX)
          AKVR = (1.0-COS(X))/(X**2)
          GO TO 100
C
C *** GENERALIZED TRAPEZOIDAL DISTRIBUTION ( ITDO = 4 )
C
 40   CONTINUE
      IF(ITDO .GT. 4) GO TO 50
        IF(AKX .LE. 0.0) AKX = 0.001
        X1 = XTM - 0.5*DXTM
        X2 = XTM + 0.5*DXTM
        A  = (1.0-ALE)/(X1+0.5)
        B  = (1.0-BTE)/(0.5-X2)
        A1 = AKX*X1
        A2 = AKX*X2
        A3 = AKX*0.5
        C1 = (ALE+0.5*A)/AKX
        C2 = A/(AKX**2)
C
        AINTR1 = C1*(SIN(A1)+SIN(A3))
     &  +C2*(COS(A1)-COS(A3)+A1*SIN(A1)-A3*SIN(A3))
        AINTI1 = C1*(COS(A3)-COS(A1))
     &  +C2*(SIN(A1)+SIN(A3)-A1*COS(A1)-A3*COS(A3))
```

```
C
         AINTR2 = (SIN(A2)-SIN(A1))/AKX
         AINTI2 = (COS(A1)-COS(A2))/AKX
C
         C3 = (1.0+X2*B)/AKX
         C4 = B/(AKX**2)
         AINTR3 = C3*(SIN(A3)-SIN(A2))
      &  -C4*(COS(A3)-COS(A2)+A3*SIN(A3)-A2*SIN(A2))
         AINTI3 = C3*(COS(A2)-COS(A3))
      &  -C4*(SIN(A3)-SIN(A2)-A3*COS(A3)+A2*COS(A2))
C
         AKVR = AINTR1+AINTR2+AINTR3
         AKVI = AINTI1+AINTI2+AINTI3
         GO TO 100
C
C *** GENERALIZED PARABOLIC DISTRIBUTION ( ITDO = 5 )
C
 50    CONTINUE
         IF(AKX .LE. 0.0) AKX = 0.001
         A  = (1.0-ALE)/((XTM+0.5)**2)
         B  = (1.0-BTE)/((0.5-XTM)**2)
         C1 = (1.0-A*XTM*XTM)/AKX
         C2 = 2.0*XTM*A/(AKX*AKX)
         C3 = A/(AKX*AKX*AKX)
         C4 = (1.0-B*XTM*XTM)/AKX
         C5 = 2.0*XTM*B/(AKX*AKX)
         C6 = B/(AKX*AKX*AKX)
         A1 = AKX*XTM
         A2 = AKX/2.0
         AKVR1 = C1*(SIN(A1)+SIN(A2))
      &  +C2*(COS(A1)-COS(A2)+A1*SIN(A1)-A2*SIN(A2))
      &  -C3*(2.0*A1*COS(A1)+2.0*A2*COS(A2)
      &  +(A1*A1-2.0)*SIN(A1)+(A2*A2-2.0)*SIN(A2))
         AKVI1 = C1*(COS(A2)-COS(A1))
      &  +C2*(SIN(A1)+SIN(A2)-A1*COS(A1)-A2*COS(A2))
      &  -C3*(2.0*A1*SIN(A1)-2.0*A2*SIN(A2)
      &  -(A1*A1-2.0)*COS(A1)+(A2*A2-2.0)*COS(A2))
         AKVR2 = C4*(SIN(A2)-SIN(A1))
      &  +C5*(COS(A2)-COS(A1)+A2*SIN(A2)-A1*SIN(A1))
      &  -C6*(2.0*A2*COS(A2)-2.0*A1*COS(A1)
      &  +(A2*A2-2.0)*SIN(A2)-(A1*A1-2.0)*SIN(A1))
         AKVI2 = C4*(COS(A1)-COS(A2))
      &  +C5*(SIN(A2)-SIN(A1)-A2*COS(A2)+A1*COS(A1))
      &  -C6*(2.0*A2*SIN(A2)-2.0*A1*SIN(A1)
      &  -(A2*A2-2.0)*COS(A2)+(A1*A1-2.0)*COS(A1))
         AKVR = AKVR1+AKVR2
         AKVI = AKVI1+AKVI2
C
 100   CONTINUE
C
       RETURN
       END
C
C ***************************************************
       SUBROUTINE AKLCAL(AKX,ILDO,XLM,AKLR,AKLI)
C ********************************************************************
C
C       THIS SUBROUTINE CALCULATES THE STEADY LOADING NOISE
```

```
C       SOURCE CHORDWISE NON-COMPACTNESS LOADING FACTOR
C                --- K-SUB-L ---
C
C ********************************************************************
C
C *** POINT-IMPULSIVE DISTRIBUTION ( ILDO = 0 )
C
      AKLR = 1.0
      AKLI = 0.0
      IF(ILDO .GT. 0) GO TO 10
        GO TO 100
C
C *** RECTANGULAR DISTRIBUTION ( ILDO = 1 )
C
 10   CONTINUE
      IF(AKX .LE. 0.0) GO TO 100
        X = ABS(0.5*AKX)
        AKLR = (SIN(X))/X
        IF(ILDO .GT. 1) GO TO 20
          GO TO 100
C
C *** LINEAR LOAD DISTRIBUTION ( ILDO = 2 )
C
 20   CONTINUE
      IF(ILDO .GT. 2) GO TO 30
        AKLI = (COS(X))/X - AKLR/X
        GO TO 100
C
C *** GENERALIZED PARABOLIC DISTRIBUTION ( ILDO = 3 )
C
 30   CONTINUE
      CALL AKVCAL(AKX,5,0.0,0.0,XLM,0.0,AKLR,AKLI)
      AKLR=1.5*AKLR
      AKLI=1.5*AKLI
C
 100  CONTINUE
C
      RETURN
      END
C
C ****************************************************
      SUBROUTINE ATMOSP(SPL,NSL,DIST,THAD,NTH)
C ********************************************************************
C
C        ATMOSPHERIC AIR ATTENUATION CORRECTIONS FOR STANDARD
C        DAY (77 DEG. F  AND 70 PCT. REL. HUM.) FROM SHIELDS
C        AND BASS CORRELATION ARE MADE TO LOSSLESS ONE-THIRD
C        OCTAVE SPECTRA
C
C ********************************************************************
C
      DIMENSION SPL(17,27),THAD(17),AA(27)
C
C *** 50 Hz - 20 KHz Atmospheric Absorption per 1000 Ft.
C
      DATA AA/
     & 0.0117,0.0186,0.0299,0.0465,0.0720,
     & 0.1163,0.1781,0.2699,0.4083,0.6111,
```

```
     & 0.8643,1.1914,1.5865,1.9902,2.4176,
     & 2.9324,3.4820,4.1945,5.2412,6.8836,
     & 9.2578,13.086,19.369,28.594,42.850,
     & 67.689,102.60/
C
      DTR = 3.1415926/180.0
C
      DO 20 J = 1,NTH
        RANGE = DIST
        IF(NSL .LT. 1) GO TO 10
          RANGE = DIST/SIN(DTR*THAD(J))
 10     CONTINUE
        DO 20 I = 1,27
          AAC = AA(I)*RANGE/1000.0
          SPL(J,I) = SPL(J,I) - AAC
          IF(SPL(J,I) .LT. 0.0) SPL(J,I) = 0.0
 20   CONTINUE
C
      RETURN
      END
C
C************************************
      SUBROUTINE BJFN(X,N,BJNX)
C***********************************************************************
C
C  SUBROUTINE FUNCTION:
C
C    COMPUTES BESSEL FUNCTIONS
C
C    INPUT:  X - ARGUMENT
C            N - ORDER
C
C    OUTPUT: BJNX - BESSEL FUNCTION
C
C **********************************************************************
C
      DIMENSION BJ(501)
C
      BJNX = 0.0
      IF(N .GE. 100)GO TO 10
        NLIM = -.0025494*X**2+1.448052*X+14.
        GO TO 20
 10   NLIM = 1.06*X+30.
 20   IF(N .GT. NLIM) RETURN
        Y = ABS(X)
        M = ABS(N)
        FM = FLOAT(M)
C
        IF(Y .LT. FM) GO TO 40
          IF(Y .GT. 8.0) GO TO 30
            Q = Y+5.0*(Y**0.33333)
            GO TO 60
 30       CONTINUE
          Q = Y+10.0
          GO TO 60
 40     CONTINUE
        IF(Y .GT. 8.0) GO TO 50
          Q = FM+5.0*(Y**0.33333)
```

```
          GO TO 60
 50     CONTINUE
        Q = FM+10.0
 60     CONTINUE
        NQ = Q
        NQ = NQ+1
        IF(NQ .GT. 500) NQ = 500
        NQM = NQ-1
        NQP = NQ+1
C
        BJ(NQP) = 0.0
        BJ(NQ ) = 1.0
C
        DO 70 I = 1,NQM
          NN = NQM-I
          IN = NN+1
          INP = IN+1
          INPP = IN+2
          BJ(IN) = (2.0*FLOAT(IN)/Y)*BJ(INP)-BJ(INPP)
 70     CONTINUE
C
        SUM = BJ(1)
        DO 80 I = 3,NQP,2
          SUM = SUM+2.0*BJ(I)
 80     CONTINUE
C
        DO 90 I = 1,NQP
          BJ(I) = BJ(I)/SUM
 90     CONTINUE
        BJNX = BJ(N+1)
C
      RETURN
      END
C
C ****************************************
      SUBROUTINE SKCAL(OM,EMR,SEARS)
C ****************************************************************
C
      PI = 3.1415926
      IF(EMR .GT. 0.8) GO TO 10
C
        BETASQ = 1.0-EMR*EMR
        OMS    = OM/BETASQ
        AMU    = EMR*OMS
        IF(AMU .GT. 1.0) GO TO 10
          BETA  = SQRT(BETASQ)
          OMK   = AMU*EMR
          SEARS = SQRT(1.0/(2.0*PI*OMS+1.0/(1.0+2.4*OMS)))
          SEARS = SEARS/BETA
          SEARS = SEARS*SQRT(1.0-(0.5*OMK)**2)
          GO TO 20
 10   CONTINUE
      EX = 2.0*EMR*OM/(1.0+EMR)
      Z  = SQRT(2.0*EX/PI)
C
      CALL FRESNL(Z,C2X,S2X)
C
      SEARS = 1.0/(PI*OM)
```

```
      SEARS = SEARS*SQRT(2.0*(C2X**2+S2X**2)/EMR)
 20   CONTINUE
      SEARS = 0.9*SEARS
C
      RETURN
      END
C
C *********************************
      SUBROUTINE FRESNL(Z,C,S)
C ***********************************************************************
C
C     FRESNEL INTEGRAL FUNCTIONS C(Z) AND S(Z)
C
C ***********************************************************************
C
      PI = 3.1415926
      X  = 0.5*PI*Z*Z
      COSX = COS(X)
      SINX = SIN(X)
C
      TOP = 1.0+0.926*Z
      BOT = 2.0+Z*(1.792+3.104*Z)
      F   = TOP/BOT
C
      TOP = 1.0
      BOT = 2.0+Z*(4.142+Z*(3.492+6.670*Z))
      G   = TOP/BOT
C
      C  =  0.5 + F*SINX - G*COSX
      S  =  0.5 - F*COSX - G*SINX
C
      RETURN
      END
C
C *******************************************
        SUBROUTINE RWVM(IZR,N,WTIV,THMOD1)
C ***********************************************************************
C
C     THIS SUBROUTINE COMPUTES THE ROTOR 1 WAKE/VORTEX CHARACTERISTICS
C     AND EVOLUTION WITH DISTANCE DOWNSTREAM.
C
C ***********************************************************************
C
        DIMENSION THETAW(300),BETA(300),WT(300),WS(300),WN(300),
     &    VT(300),VN(300),VS(300),ALPHA(300),VTP(300),VPS(300),
     &    WT0(300),WTIF(300),WSC(300),WNC(300),PHI(10),WTOT(300)
C
        COMMON /RWAKE1/ ISHAPE,BETAW,NWHM,VREF,IWAKE
C
        COMMON /PRINTD/ IPRT,IPR,IPRNTW
C
        COMMON /COM12/ NZ,DTR,RTD,VO,CO,SXOCH(51)
C
        COMMON /HMONIC/ ST(300),VPN(300),FCA(21),FCB(21),FCDB(21)
C
        COMMON /FANVTX/ NSTR,SSIGR(51),SSEMA(51),SSTHET(51),SSEMT(51)
C
        COMMON /BLADE1/ THETA1(51), SIGR1(51), DPINT1(51), CHORD1(51),
```

```
     &    TMOC1(51), Z1(51), ZMC1(51), YMC1(51), BETAP1(51), MCA1(51),
     &    FA1(51), BETPP1(51), DELBP1, SCD1(51), EMT1, NB1
C
        COMMON /BLADE2/ THETA2(51), SIGR2(51) ,DPINT2(51), CHORD2(51),
     &    TMOC2(51), Z2(51), ZMC2(51), YMC2(51), BETAP2(51), MCA2(51),
     &    FA2(51), BETPP2(51), DELBP2, SCD2(51), EMT2, NB2
C
        COMMON /VTEX1/ ITPVTX,IHBVTX,TAU,ALPHR,CHORD,WT00,B1
C
        COMMON /VTEX2/ SAODS(2),SCIRC0(2),SO0(2),SVSDV0(2),SCL(2),
     &    SFRL(2)
C
        COMMON /VTEX3/ SBN(2),SBR(2),SDIST,RAWDS,R,VINRVT(300,2),
     &    VISRVT(300,2),HTR,VINRV(300),VISRV(300)
C
        COMMON /VTEX4/ CLAV,CAV,WT00T,ABR(51),CI,TVTI
C
C ***     CI=CIRCULATION INDEX FOR TIP VORTEX
C ***      TVTI=TIP VORTEX TRAJECTORY INDEX
C
        REAL MCA1,MCA2
C
        PI = 4.*ATAN(1.)
        B1 = NB1
        B2 = NB2
        IF(IPRNTW .GT.0) THEN
            IPR = 0
            DO 10 IP = 0,11
                IF(IZR .EQ. 1+IP*IPRT) THEN
                    IPR = 1
                    WRITE (16,1000)
                    WRITE (16,1010) IZR
                    GO TO 20
                END IF
 10         CONTINUE
        END IF
 20     CONTINUE
C
        VVTR   = EMT2/EMT1
        FOPT   = 1.0
        SADIN  = 90.0-BETPP1(IZR)
        SADINS = 90.0-BETPP2(IZR)
        SEMA   = VO/CO
        SIGS1  = SIGR2(IZR)
        SEMT   = EMT1*Z1(IZR)
        VWHEEL = SEMT*CO
        R      = Z1(IZR)
        SIG1   = SIGR1(IZR)
        SXOCH1 = SXOCH(IZR)
        STHETA = THETA1(IZR)*THMOD1
        ISTR   = IZR
        CD     = SCD1(IZR)
        IF(IPR.EQ.1 .AND. IPRNTW.GT.0) WRITE(16,1020) CD
C
C      ==============================================
C
C      SIMILARITY AND CORRELATION CALCULATIONS
C
```

```
C      ==============================================
C
C  ***   WAKE PARAMETERS
C
        BETA0 = ATAN2((1.-STHETA)*SEMT,SEMA)
        ALPHR = BETA0
        ALPHS = ATAN2(STHETA*SEMT,SEMA)
C
C       CHECK FOR VORTICES
C
        IF(ITPVTX.EQ.0.AND.IHBVTX.EQ.0) GO TO 30
         IF(IZR.NE.1) GO TO 40
          SIG1T    = SIG1
          SEMAT    = SEMA
          STHETAT  = STHETA
          SEMTT    = SEMT
C
          CALL VORTX1(SIG1T,SEMAT,STHETAT,SEMTT)
C
          GO TO 40
 30     IF(IZR.EQ.1) WRITE(06,1030)
C
C ***    W(N) EVALUATED AT STATOR 1/4 CHORD POINT
C
 40     SXOCH1  = SXOCH1+0.25*B1/B2*SIGS1/SIG1
        WT00    = SEMA/(SEMT*COS(BETA0))
        BNB     = NB1
C
        SDIST   = SXOCH1/COS(BETA0)
C
        CALL WAKE1 (IWAKE,CD,SDIST,WT00,ALPHR,BETA0,SIG1,WTDC,DL0)
C
        IF(IPR.NE.1) GO TO 50
         WRITE (16,1040) WT00
         WRITE (16,1050) WTDC
         WRITE (16,1060) SXOCH(IZR)
         WRITE (16,1070) SDIST
         WRITE (16,1080) DL0
 50     CONTINUE
C
C      =================================================
C
C      DETERMINE RELATIVE,ABSOLUTE,AND PERTURBATION VELOCITIES
C
C      =================================================
C
C ***  SPECIFY ANALYSIS TRAVERSE DISTANCE STEP SIZE
C
        RN          = FLOAT(N)
        NPS         = (N-1)/2
        S           = 2.*PI/BNB
        NI          = N-1
        RNI         = FLOAT(NI)
        SI          = S/RNI
        RNPS        = FLOAT(NPS)
        THETAW(1)   = -RNPS*SI
        ST(1)       = THETAW(1)/(2.*ABS(THETAW(1)))
C
```

```
      DO 60 I=2,N
          THETAW(I) = THETAW(I-1)+SI
          ST(I)     = THETAW(I)/(2.*ABS(THETAW(1)))
60    CONTINUE
C
C *** DETERMINE INVISCID DISTRIBUTION OF WT
C
      BETA0    = BETA0*DTR
      WTINC    = WTIV/RNI
      WT0(1)   = WT00
      WTIF(1)  = 1.0
      ND2      = (N+1)/2
      N22      = ND2+1
      DO 70 I=2,ND2
          WTIF(I) = WTIF(I-1)-WTINC
70    CONTINUE
      DO 80 I=N22,N
          WTIF(I) = WTIF(1)+WTIV/2.+(N22-I)*WTINC
80    CONTINUE
C
C *** DETERMINE TANGENTIAL ANGLE DISTRIBUTION
C
      BETAW    = BETAW*DTR
      BETA(1)  = BETA0
      DO 90 I=2,N
          BETA(I) = BETA(1)
90    CONTINUE
C
      IMIDDL = (N+1)/2
C
C *** SPECIFY EXIT-FLOW TOTAL RELATIVE VELOCITY PROFILE
C
      DO 100 I=1,IMIDDL
          DLSPP = DL0*S
          DLSOP = DLSPP/2.
          DL    = DLSOP
          DT    = THETAW(I)/DL
          DT1   = (THETAW(I)+S)/DL
C
          CALL WAKE2 (IWAKE,ISHAPE,DT,PP)
C
          WT(I)    = WTDC*(1.-PP)
          WT(I)    = WT(I)*WTIF(I)+WT0(1)-WTDC
          II       = I
          BETA(I)  = BETA(I)+BETAW*EXP(PP)
C
          CALL WAKE2 (IWAKE,ISHAPE,DT1,PP1)
C
          PPT      = PP+PP1
          WTOT(I)  = WTDC*(1.-PPT)
          WTOT(I)  = WTOT(I)*WTIF(I)+WT0(1)-WTDC
100   CONTINUE
C
C *** CALCULATE RELATIVE VELOCITY COMPONENTS
C
      IMIDL1 = IMIDDL+1
      DEL    = WTOT(IMIDDL)-WT(IMIDDL)
C
```

```
      DO 110 I=IMIDL1,N
         DT  = THETAW(I)/DL
         DT1 = (THETAW(I)-S)/DL
C
         CALL WAKE2 (IWAKE,ISHAPE,DT,PP)
C
         WT(I)    = WTDC*(1.-PP)
         WT(I)    = WT(I)*WTIF(I)+WT0(1)-WTDC
         BETA(I)  = BETA(I)+BETAW*EXP(PP)
C
         CALL WAKE2 (IWAKE,ISHAPE,DT1,PP1)
C
         PPT      = PP+PP1
         WTOT(I)  = WTDC*(1.-PPT)
         WTOT(I)  = WTOT(I)*WTIF(I)+WT0(1)-WTDC
         II       = I
 110  CONTINUE
C
      DO 120 I=1,N
         WTOT(I)  = WTOT(I)-DEL
         BETDEL   = BETA(I)-BETA(1)
         WS(I)    = WTOT(I)*COS(BETDEL)
         WN(I)    = WTOT(I)*SIN(BETDEL)
         II       = I
 120  CONTINUE
C
      IF (ITPVTX.EQ.0.AND.IHBVTX.EQ.0) GO TO 130
C
C *** CALCULATE VORTEX INDUCED VELOCITY FIELD
C
      NST   = N
      RAWDS = B1/(2.*PI)
C
      CALL VORTX2(ISTR,NST,SIG1T,SEMAT,STHETAT,SEMTT,SEMT,SIG1)
C
C
C *** INTEGRATE WAKE AND VORTEX GUST DESCRIPTIONS
C
 130  N0 = (N-1)/2
      N1 = (N+1)/2
C
      DO 140 I=1,N1
         IF (ITPVTX.EQ.0.AND.IHBVTX.EQ.0) VISRV(I) = 0.
         IF (ITPVTX.EQ.0.AND.IHBVTX.EQ.0) VINRV(I) = 0.
         WSC(I) = VISRV(I)+WS(N0+I)
         WNC(I) = VINRV(I)+WN(N0+I)
 140  CONTINUE
C
      N2 = 2*N0
C
      DO 150 I=N1,N
         IF (ITPVTX.EQ.0.AND.IHBVTX.EQ.0) VISRV(I) = 0.
         IF (ITPVTX.EQ.0.AND.IHBVTX.EQ.0) VINRV(I) = 0.
         WSC(I) = VISRV(I)+WS(I-N0)
         WNC(I) = VINRV(I)+WN(I-N0)
 150  CONTINUE
C
      N3 = 2*N-1
```

```
C
        DO 160 I=N,N3
            WSC(I) = WSC(I-N2)
            WNC(I) = WNC(I-N2)
 160    CONTINUE
C
        DO 170 I=1,N3
            WS(I) = WSC(I)
            WN(I) = WNC(I)
            WT(I) = (WSC(I)**2+WNC(I)**2)**0.5
            AAA    = WS(I)/WT(I)
            IF (AAA.GE.1.0) AAA = 1.0
            BETA(I) = ACOS(AAA)+BETA0
            RLI      = FLOAT(I)
            RLN3     = FLOAT(N3)
            ST(I)    = (RLI-1.)/(RLN3-1.)
            ST(I)    = ST(I)*2.
 170    CONTINUE
C
        DO 180 I=N,N3
            THETAW(I) = THETAW(I-N2)
 180    CONTINUE
C
        WTHN1     = WT(N1)*SIN(BETA(N1))
        WXN1      = WT(N1)*COS(BETA(N1))
        VXN1      = WXN1
        VTHN1     = 1.+VVTR-WTHN1
        VTN1      = (VXN1**2+VTHN1**2)**0.5
        ALPHA(N1) = ACOS(VXN1/VTN1)
        ALPDN1    = ALPHA(N1)-ALPHA(N1)
        VS(N1)    = VTN1*COS(ALPDN1)
C
C ***  DETERMINE ABSOLUTE VELOCITY PROFILE
C
C
        DO 190 I=1,N3
            WTH      = WT(I)*SIN(BETA(I))
            WX       = WT(I)*COS(BETA(I))
            VX       = WX
            VTH      = 1.0+VVTR-WTH
            VT(I)    = (VX**2+VTH**2)**0.5
            ALPHA(I) = ACOS(VX/VT(I))
C
C ***  DETERMINE ABSOLUTE VELOCITY COMPONETS
C
            ALPDEL = ALPHA(I)-ALPHA(N1)
            VS(I)  = VT(I)*COS(ALPDEL)
            VN(I)  = VT(I)*SIN(ALPDEL)
C
C ***  DETERMINE TOTAL PERTURBATION VELOCITY
C
            VPN(I) = VN(I)
            VPS(I) = VS(I)-VS(N1)
            VTP(I) = (VPS(I)**2+VPN(I)**2)**0.5
C
 190    CONTINUE
C
        DO 195 I=1,N
```

```
            VINRV(I) = 0.
            VISRV(I) = 0.
195     CONTINUE
C     ======================================================
C
C     PRINT NUMERICAL VELOCITY PROFILES
C
C     ======================================================
C
        DO 200 I=1,N3
            THETAW(I) = THETAW(I)*RTD
            BETA(I)   = BETA(I)*RTD
            ALPHA(I)  = ALPHA(I)*RTD
 200    CONTINUE
C
        IF (IPRNTW.LE.0) GO TO 250
C
        IF(IPR.NE.1) GO TO 240
        WRITE (16,1090)
        WRITE (16,1100)
        DO 210 I=1,N3,10
            WRITE (16,1110) I,THETAW(I),ST(I),WT(I),WS(I),WN(I),BETA(I)
 210    CONTINUE
C
        WRITE (16,1120)
        WRITE (16,1130)
        DO 220 I=1,N3,10
            WRITE (16,1110) I,THETAW(I),ST(I),VT(I),VS(I),VN(I),ALPHA(I)
 220    CONTINUE
C
        WRITE (16,1140)
        WRITE (16,1150)
        DO 230 I=1,N3,10
            WRITE (16,1160) I,THETAW(I),ST(I),VTP(I),VPS(I),VPN(I)
 230    CONTINUE
 240    CONTINUE
C
C
C     ==============================================================
C
C     CALCULATE HARMONIC CONTENT OF ROTOR EXIT FLOW
C
C     ==============================================================
C
 250    IF (FOPT.NE.1.0) GO TO 260
C
        CALL HRMNIC (ISTR,N,NWHM,VREF,VWHEEL)
C
 260    CONTINUE
C
C     FORMAT STATEMENTS
C
 1000 FORMAT (///18X,'ROTOR WAKE/VORTEX FLOW PROGRAM')
 1010 FORMAT (////2X,'--- STREAMLINE NUMBER',I3,' ---')
 1020 FORMAT (/5X,'CD =',F8.4)
 1030 FORMAT (/2X,'TIP VORTEX NOT INCLUDED')
 1040 FORMAT (5X,'WFS/UT',5X,'=',F8.4)
 1050 FORMAT (5X,'WD/UT',6X,'=',F8.4)
```

```
 1060 FORMAT(5X,'X/C',8X,'=',F8.4)
 1070 FORMAT (5X,'SDIST',6X,'=',F8.4)
 1080 FORMAT (5X,'WAKE WIDTH',1X,'=',F8.4)
 1090 FORMAT (/5X,'VELOCITY PROFILE RELATIVE TO FORWARD ROTOR')
 1100 FORMAT (/9X,'I',5X,'THETAW',5X,'S',8X,'WT',7X,'WS',7X,
     1 'WN',7X,'BETA')
 1110 FORMAT (7X,I4,F9.2,4F9.5,F11.5)
 1120 FORMAT (/5X,'VELOCITY PROFILE RELATIVE TO AFT ROTOR')
 1130 FORMAT (/9X,'I',5X,'THETAW',5X,'S',8X,'VT',7X,'VS',7X,'VN',
     1 6X,'ALPHA')
 1140 FORMAT (/5X,'PERTURBATION VELOCITY PROFILE RELATIVE TO AFT ROTOR'
     1)
 1150 FORMAT (/9X,'I',5X,'THETAW',5X,'S',7X,'VPT',6X,'VPS',6X,
     1 'VPN')
 1160 FORMAT (7X,I4,F9.2,4F9.5)
C
        RETURN
        END
C
C **********************************************************************
      SUBROUTINE WAKE1 (IWAKE,CD,SDIST,WT00,ALPHR,BETA0,SIG1,WTDC,DL0)
C **********************************************************************
C
C     SUBROUTINE TO CALCULATE WAKE CENTERLINE DEFECT AND SEMI-WAKE WIDTH
C
C **********************************************************************
C
      PI = 4.*ATAN(1.)
      GO TO (100,200,300),IWAKE
C
C *** LINEAR RATIONAL FUNCTION FOR ROTOR WAKE PROFILE
C
 100  CONTINUE
        CDEXP1 = CD**(0.125)
        CDEXP2 = CD**(0.25)
        IF(SIG1 .GE. 1.0) GO TO 110
          DL0C = ((0.2375*SDIST*CDEXP1+0.034125)/
     &    (0.357*SDIST*CDEXP1+1.0))
          DL0  = DL0C*SIG1
          GO TO 120
 110    CONTINUE
        DL0  = ((0.31875*SDIST*CDEXP1+0.048)/
     &  (0.268125*SDIST*CDEXP1+1.0))
 120    CONTINUE
        WTDC = CDEXP2*((0.3675*SDIST+1.95)/(7.65*SDIST+1.0))
        GO TO 400
C
C *** SILVERSTEIN/ KEMP & SEARS MODEL FOR ROTOR WAKE
C
 200  CONTINUE
        WTDC = SQRT(CD)*(1.21*(SDIST+.3)**(-1.))
        DL0  = SQRT(CD)*SIG1*(0.68*(SDIST+.15)**.5)
        GO TO 400
C
C *** MUGRIDGE & MORFEY MODEL FOR ROTOR WAKE
C
 300  CONTINUE
        A1A  = 2.0
```

```
        WTDC = SQRT(CD)*A1A/SQRT(2.)*((SDIST+2.*CD)**(-.5))
        DL0  = CD*SIG1*(.5+EXP(-.16/CD*SDIST))*(WTDC)**(-1.0)
C
 400  CONTINUE
      IF (IWAKE .GT. 1) DL0 = DL0/COS(ALPHR)
      WTDC = WTDC*WT00
      IF (DL0 .GE. 0.5) DL0 = 0.5
      BETA0 = BETA0*180./PI
C
      RETURN
      END
C
C **********************************************
      SUBROUTINE WAKE2 (IWAKE,ISHAPE,DTIN,PP)
C ********************************************************************
C
C     SUBROUTINE TO CALCULATE TANGENTIAL WAKE PROFILE
C
C ********************************************************************
C
      DT = DTIN
      PI = 4.*ATAN(1.)
C
      IF(ISHAPE .NE.2) GO TO 10
        IF(DT .GT.  10.0) DT =  10.0
        IF(DT .LT. -10.0) DT = -10.0
 10   CONTINUE
C
      GO TO (100,200,300),IWAKE
C
C *** LINEAR RATIONAL FUNCTION FOR ROTOR WAKE PROFILE
C
 100  CONTINUE
        GO TO (110,120),ISHAPE
 110    PP = 2./(EXP(1.3169579*DT)+EXP(-1.3169579*DT))
        GO TO 400
 120    PP = EXP(-0.6931472*DT*DT)
        GO TO 400
C
C *** SILVERSTEIN/KEMP & SEARS WAKE PROFILE
C
 200  CONTINUE
        GO TO (210,220),ISHAPE
 210    PP = 2./(EXP(1.3169579*DT)+EXP(-1.3169579*DT))
        GO TO 400
 220    PP = EXP(-PI*DT*DT/4.)
        GO TO 400
C
C *** MUGRIDGE & MORFEY WAKE PROFILE
C
 300  CONTINUE
        GO TO (310,320),ISHAPE
 310    PP = 2./(EXP(1.3169579*DT)+EXP(-1.3169579*DT))
        GO TO 400
 320    PP = EXP(-PI*DT*DT/4.)
C
 400  CONTINUE
      IF (PP .LT. 1.E-20) PP = 1.E-20
```

```
C
      RETURN
      END
C
C ****************************************************
        SUBROUTINE VORTX1(SIG1T,SEMAT,STHETAT,SEMTT)
C *****************************************************************
C     SUBROUTINE VORTX1 COMPUTES VORTEX STRENGTH AND RADIUS OF
C     TIP AND HUB VORTICES
C *****************************************************************
C
        COMMON /FANVTX/ NSTR,SSIGR(51),SSEMA(51),SSTHET(51),SSEMT(51)
C
        COMMON /VTEX1/ ITPVTX,IHBVTX,TAU,ALPHR,CHORD,WT00,B1
C
        COMMON /VTEX2/ SAODS(2),SCIRC0(2),SO0(2),SVSDV0(2),SCL(2),
     &    SFRL(2)
C
        COMMON /VTEX4/ CLAV,CAV,WT00T,ABR(51),CI,TVTI
C
        COMMON /PRINTD/IPRT,IPR,IPRNTW
C
        PI=4.0*ATAN(1.)
C
        IF (ITPVTX.EQ.0) GO TO 10
C
        SIGR      = SIG1T
        SEMA      = SEMAT
        STHETA    = STHETAT
        SEMT      = SEMTT
        BETA0     = ATAN((1.-STHETA)*SEMT/SEMA)
        ALPHR     = BETA0
        CHORD     = SIGR
        WT00T     = SEMA/(SEMT*COS(BETA0))
        SEMR      = SEMA/COS(ALPHR)
        VTHEM0    = 0.5586*WT00T*SQRT(CLAV)
        AODST     = 0.0014*CAV*SQRT(CLAV)*B1/PI
        CIRC0T    = 2.0*PI*VTHEM0*AODST
        O0T       = CIRC0T/(AODST*2.*PI)
        VSDV0T    = 0.0
        SAODS(1)  = AODST
        SCIRC0(1) = CIRC0T
        SO0(1)    = O0T
        SVSDV0(1) = VSDV0T
        SCL(1)    = CLT
        SFRL(1)   = FRLT
        GO TO 20
C
 10     SAODS(1)  = 0.0
        SCIRC0(1) = 0.0
        SO0(1)    = 0.0
        SVSDV0(1) = 0.0
        SCL(1)    = 0.0
        SFRL(1)   = 0.0
        IF(IPR.EQ.1 .AND. IPRNTW.EQ.1) WRITE (16,1000)
C
 20     IF (IHBVTX.EQ.0) GO TO 30
        SIGR      = SSIGR(NSTR)
```

```
        SEMA      = SSEMA(NSTR)
        STHETA    = SSTHET(NSTR)
        SEMT      = SSEMT(NSTR)
        BETA0     = ATAN((1.-STHETA)*SEMT/SEMA)
        ALPHR     = BETA0
        CHORD     = SIGR
        WT00T     = SEMA/(SEMT*COS(BETA0))
        FRLH      = 0.2
        SEMR      = SEMA/COS(ALPHR)
        CLH       = 2.0*SEMT*STHETA/(SEMR*SIGR)
        AODSH     = 0.2*SIGR
        CIRC0H    = FRLH*CLH/2.*CHORD*WT00T
        O0H       = CIRC0H/(AODSH*2.*PI)
        VSDV0H    = 0.2
        SAODS(2)  = AODSH
        SCIRC0(2) = CIRC0H
        SO0(2)    = O0H
        SVSDV0(2) = VSDV0H
        SCL(2)    = CLH
        SFRL(2)   = FRLH
        GO TO 40
C
 30     SAODS(2)  = 0.0
        SCIRC0(2) = 0.0
        SO0(2)    = 0.0
        SVSDV0(2) = 0.0
        SCL(2)    = 0.0
        SFRL(2)   = 0.0
        IF(IPR.EQ.1 .AND. IPRNTW.EQ.1) WRITE (16,1010)
C
 40     CONTINUE
C
C     FORMAT STATEMENTS
C
 1000 FORMAT(//5X,'NO TIP VORTEX')
 1010 FORMAT(//5X,'NO HUB VORTEX')
C
        RETURN
C
C     END OF SUBROUTINE VORTX1
        END
C
C *****************************************************************
        SUBROUTINE VORTX2 (IS,N,SIG1T,SEMAT,STHETAT,SEMTT,SEMT,SIG1)
C *****************************************************************
C
C     SUBROUTINE VORTX2 COMPUTES THE VELOCITY FIELD INDUCED BY
C     TIP AND HUB VORTICES AT ALL RADIAL LOCATIONS
C *****************************************************************
C
        COMMON /VTEX1/ ITPVTX,IHBVTX,TAU,ALPHR,CHORD,WT00,B1
C
        COMMON /VTEX2/ SAODS(2),SCIRC0(2),SO0(2),SVSDV0(2),SCL(2),
     &    SFRL(2)
C
        COMMON /VTEX3/ SBN(2),SBR(2),SDIST,RAWDS,R,VINRVT(300,2),
     &    VISRVT(300,2),HTR,VINRV(300),VISRV(300)
C
```

```
      COMMON /VTEX4/ CLAV,CAV,WT00T,ABR(51),CI,TVTI
C
      COMMON /PRINTD/ IPRT,IPR,IPRNTW
C
      DIMENSION ANL(300),SYO(2)
C
      PI=4.0*ATAN(1.0)
C
      IF (ITPVTX.EQ.0.OR.IHBVTX.EQ.0) GO TO 10
      NVORTX = 2
      SYO(1) = 0.0
      SYO(2) = RAWDS-HTR*B1/(2.*PI)
      GO TO 30
C
 10   NVORTX=1
      IF (ITPVTX.EQ.0) GO TO 20
      SYO(1)=0.0
      SYO(2)=0.0
      GO TO 30
C
 20   SYO(1)    = RAWDS-HTR*B1/(2.*PI)
      SYO(2)    = 0.0
      SAODS(1)  = SAODS(2)
      SCIRC0(1) = SCIRC0(2)
      SO0(1)    = SO0(2)
      SVSDV0(1) = SVSDV0(2)
      SCL(1)    = SCL(2)
      SFRL(1)   = SFRL(2)
      SBN(1)    = SBN(2)
C
 30   CONTINUE
C
      DO 130 IVORTX=1,NVORTX
          AODS   = SAODS(IVORTX)
          CIRC0  = SCIRC0(IVORTX)
          O0     = SO0(IVORTX)
          VSDV0  = SVSDV0(IVORTX)
          CL     = SCL(IVORTX)
          FRL    = SFRL(IVORTX)
          BN     = SBN(IVORTX)
          YO     = SYO(IVORTX)
          SDISTV = SDIST*CHORD/(CAV*B1/PI)
          A      = (0.01584*SDISTV+0.0014)/(0.184*SDISTV+1.0)*
     &                    CAV*SQRT(CLAV)*B1/PI
          VTHEM  = (0.024*SDISTV+0.5586)/(0.0504*SDISTV+1.0)*
     &                    WT00T*SQRT(CLAV)
          CIRC   = 2.0*PI*VTHEM*A
          CIRC   = CI*CIRC
          CIRC   = CIRC/(1.+SDISTV)**0.25
          VSDV   = VSDV0
C
          IF (ITPVTX.EQ.1.AND.IHBVTX.EQ.1) GO TO 50
          IF (ITPVTX.EQ.0) GO TO 40
C
          ZQRT   = SDIST*COS(ALPHR)*R*SIG1*2.*PI/B1
          BRRT   = TVTI*ZQRT/(16.0*ZQRT+1.0)
          BR     = BRRT*B1/(2.*PI)
          BR     = BR+TAU*SIG1T
```

```
            SBR(1) = BR
            IF(IPR.EQ.1 .AND. IPRNTW.EQ.1) WRITE (16,1000)
            GO TO 70
C
 40         BR     = YO-A
            SBR(1) = BR
            IF(IPR.EQ.1 .AND. IPRNTW.EQ.1) WRITE (16,1010)
            GO TO 70
C
 50         IF (IVORTX.EQ.1) GO TO 60
            BR     = YO-A
            SBR(2) = BR
            IF(IPR.EQ.1 .AND. IPRNTW.EQ.1) WRITE (16,1010)
            GO TO 70
C
 60         BR     = A+TAU*SIGR
            SBR(1) = BR
            IF(IPR.EQ.1 .AND. IPRNTW.EQ.1) WRITE (16,1000)
C
 70         CONTINUE
C
            IF(IPR.NE.1 .OR. IPRNTW.NE.1) GO TO 80
            WRITE (16,1020) AODS
            WRITE (16,1030) A
            WRITE (16,1040) VSDV0
            WRITE (16,1050) VSDV
            WRITE (16,1060) CIRC0
            WRITE (16,1070) CIRC
            WRITE (16,1080) BN
            WRITE (16,1090) BRRT
C
 80         CONTINUE
C
            RN     = FLOAT(N)
            RNI    = RN-1.0
            SANL   = 1./RNI
            ANL(1) = 0.0
            DO 90 I=2,N
                ANL(I) = ANL(I-1)+SANL
 90         CONTINUE
C
            ARL = RAWDS*(1.-R)
            ARL = ARL+TAU*SIG1T
C
C       FORCED OR FREE VORTEX?
C
            DO 120 I=1,N
                RNVL = BN-ANL(I)
                RRVL = BR-ARL
                RAL  = (RNVL**2+RRVL**2)**0.5
                IF (RAL.LT.A) GO TO 100
C
C     FREE VORTEX NORMAL VELOCITY CALCULATION
C
                VM  = 2.*PI*(ARL-YO-BR)
                VNN = 2.*PI*(ARL-YO+BR)
                XN  = 2.*PI*(ANL(I)-BN)
                D1  = COSH(VM)-COS(XN)
```

```
                D2   = COSH(VNN)-COS(XN)
C
                VINRVT(I,IVORTX) = -CIRC/2.0*(SINH(VM)/D1-SINH(VNN)/D2)
                VINRVT(I,IVORTX) = VINRVT(I,IVORTX)*SEMTT/SEMT
C
                GO TO 110
C
C     FORCED VORTEX NORMAL VELOCITY CALCULATIOIN
C
 100            IF (RNVL.GT.RAL) RNVL = RAL
                THETAV = ACOS(RNVL/RAL)
                X0     = A*COS(THETAV)
                Y0     = A*SIN(THETAV)
                IF (ARL.GT.BR) Y1 = Y0+BR
                IF (ARL.LE.BR) Y1 = BR-Y0
                X1     = BN-X0
C
                VM     = 2.*PI*(Y1-YO-BR)
                VNN    = 2.*PI*(Y1-YO+BR)
                XN     = 2.*PI*(X1-BN)
                D1     = COSH(VM)-COS(XN)
                D2     = COSH(VNN)-COS(XN)
                VINPRF = -CIRC/2.0*(SINH(VM)/D1-SINH(VNN)/D2)
C
                VINRVT(I,IVORTX) = (RAL/A)*VINPRF
                VINRVT(I,IVORTX) = VINRVT(I,IVORTX)*SEMTT/SEMT
C
 110            CONTINUE
C
C
C     VORTEX STREAMWISE VELOCITY CALCULATION
C
                PP = -0.693*((RAL/A)**2)
                IF (PP.LT.-40.) PP = -40.
C
                VISRVT(I,IVORTX) = -VSDV*EXP(PP)
                VISRVT(I,IVORTX) = VISRVT(I,IVORTX)*SEMTT/SEMT
C
 120         CONTINUE
 130       CONTINUE
C
           DO 140 I=1,N
               DO 140 IVORTX=1,NVORTX
               VINRV(I) = VINRV(I)+VINRVT(I,IVORTX)
               VISRV(I) = VISRV(I)+VISRVT(I,IVORTX)
 140       CONTINUE
C
C     FORMAT STATEMENTS
C
 1000 FORMAT (///2X,'TIP VORTEX PARAMETERS')
 1010 FORMAT (///2X,'HUB VORTEX PARAMETERS')
 1020 FORMAT (5X,'A0/S',8X,'=',F8.4)
 1030 FORMAT (5X,'A/S',9X,'=',F8.4)
 1040 FORMAT (5X,'VSDV0/U',5X,'=',F8.4)
 1050 FORMAT (5X,'VSDV/U',6X,'=',F8.4)
 1060 FORMAT (5X,'CIRC0/(S*U)',1X,'=',F8.4)
 1070 FORMAT (5X,'CIR/(S*U)',3X,'=',F8.4)
 1080 FORMAT (5X,'BN',10X,'='F8.4)
```

```
 1090 FORMAT (5X,'BRRT',10X,'='F8.4)
C
        RETURN
C
C     END OF SUBROUTINE VORTX2
        END
C
C *****************************************************
      SUBROUTINE HRMNIC (ISTR,N,NWHM,VREF,VWHEEL)
C **********************************************************************
C
C     COMPUTES HARMONIC CONTENT OF ROTOR WAKE FLOW AT
C     1/4 CHORD POINT OF STATOR
C
C **********************************************************************
C
      COMMON /PRINTD/ IPRT,IPR,IPRNTW
C
      COMMON /HMONIC/ ST(300),VPN(300),FCA(21),FCB(21),FCDB(21)
C
      COMPLEX CSUM,CPWR,CERC
C
      PI = 4.*ATAN(1.)
      RN = FLOAT(N)
      NS = (N+1)/2
      NF = NS+N-1
C
      DO 100 I = NS,NF
        ST(I) = ST(I)-0.5
 100  CONTINUE
C
      IF(IPR .NE. 1) GO TO 110
        IF(IPRNTW .LE. 0) GO TO 110
          WRITE (16,1000)
          WRITE (16,1020)
 110  CONTINUE
C
      DO 220 NH = 1,NWHM
        RNH  = FLOAT(NH)
        CSUM = CMPLX(0.0,0.0)
        HD3  = (ST(NF)-ST(NS))/(3.*(RN-1.))
        ICH  = 1
C
C *** NUMERICAL INTEGRATION - SIMPSON'S RULE
C
        DO 210 I = NS,NF
          ICH  = ICH*(-1)
          CPOE = RNH*PI*ST(I)/0.5
          CPWR = CMPLX(0.0,CPOE)
          CERC = CEXP(CPWR)
          IF (I .EQ. NS .OR. I .EQ. NF) GO TO 200
            CSUM = CSUM+VPN(I)*CERC*(3+ICH)
            GO TO 210
 200      CONTINUE
          CSUM = CSUM+VPN(I)*CERC
 210    CONTINUE
C
        CSUM     = CSUM*HD3
```

```
        CMOD      = CABS(CSUM)
        ARPC      = REAL(CSUM)
        BIPC      = AIMAG(CSUM)
        PHASE     = ATAN(BIPC/ARPC)
        FCA(NH)   = 2.*CMOD
        FCB(NH)   = PHASE
        FCDB(NH)  = 20.0*ALOG10(ABS(FCA(NH)))
        FCDB(NH)  = FCDB(NH)+20.0*ALOG10(VWHEEL/VREF)
C
        IF(IPR.EQ.1.AND.IPRNTW.GT.0)
     &  WRITE (16,1030) NH,FCA(NH),FCB(NH),FCDB(NH)
 220  CONTINUE
C
C *** RESCALE INTEGRATION INTERVAL
C
      DO 300 I = NS,NF
        ST(I) = ST(I)+0.5
 300  CONTINUE
C
C *** FORMAT STATEMENTS ------------------------------------------------
C
 1000 FORMAT (///2X,'HARMONIC CONTENT OF ROTOR WAKE FLOW')
 1020 FORMAT (/8X,'NH',3X,'2*CMOD',4X,'PHASE',4X,
     & '20*LOG(2*CMOD), DB')
 1030 FORMAT (7X,I3,F10.6,F11.6,F14.2)
C
      RETURN
      END
C **********************************************************************
C *                          MAIN PROGRAM                              *
C *      COUNTER  ROTATION PROPELLER INSTALLATION EFFECTS - 'CRPIE'    *
C **********************************************************************
C
C           THIS PROGRAM  CALCULATES 'INSTALLATION EFFECTS'
C               ON COUNTER ROTATION PROPELLER NOISE.
C
C        IT REQUIRES ACCESS TO THE 'IMSL' LIBRARY PROGRAMS:
C                ' CERFE, FFTRF, NEQNF, U4INF, UMINF '.
C
C **********************************************************************
C
        SUBROUTINE CRPIE(ISOLAT)
C
      DIMENSION ALD(51),CD(51),CL(51)
C
      DIMENSION ALDA(51),CDA(51),CLA(51)
C
      DIMENSION X(3),PSPANV(51)
C
      DIMENSION FUFV1(5),TF1V(5),TF2V(5),
     &          QF1V(5),QF2V(5),
     &          BSWPFD(51),BSWPAD(51)
C
      DIMENSION RADV(51),RADVA(51),CHDV(51),CHDVA(51),BETPV1(51),
     &          BETPV2(51),SIGV(51),SIGVA(51),
     &          FRV(51),FRVA(51),FRVAN(51),YSIGV(51),
     &          YSIGVA(51),U(1),S(1),POWF(51),
```

```
     &           POWA(51),FRDIM(51),YBETP(51),
     &           YBETPA(51),XV(2),XGUESS(2),TMOCV(51),TMOCV2(51)
C
      DIMENSION XG1(2),XSCALE(2),IPARAM(7),RPARAM(7)
C
      DIMENSION THRFRV(51),THRRRV(51)
C
      DIMENSION YMCV1(51),YMCV2(51),ZMCV1(51),ZMCV2(51),
     &           CHD1(51),CHD2(51)
C
      DIMENSION YMCD1(51),YMCD2(51),ZMCD1(51),ZMCD2(51),
     &           CHDD1(51),CHDD2(51),XQCVD(51),XQCVDA(51)
C
      DIMENSION RADDES(51),FRDES(51),YSGVD(51),YSGVDA(51),
     &           YBPD(51),YBPDA(51),THSRP(2,51),
     &           SIGVT(51),THSRPT(51),TTRT(51),QTRT(51),TDUT(51),
     &           RLV(51),XRVV(2,51),XRV(51),
     &           QDUT(51),QDVT(51),THCKV(51),THTDIV(20,2),THKVIF(51),
     &           THKVIR(51)
C
      DIMENSION TTR(2,51),QTR(2,51),TDUTR(2,51),TDVTR(2,51),
     &           QDUTR(2,51),QDVTR(2,51),TDVT(51)
C
      DIMENSION RADP(51),XLEP(51),XTEP(51),CAMPD(51),FRADP(51),
     &           XLEPD(51),XTEPD(51),CAMPDD(51)
C
      REAL       LAM,LAMA,LAM0,LAM2,MFLT,MTC
C
      COMMON /A/ ALD,CD,CL,LAM,FR,SIG,THETDF,DEGR,NPF,
     &             ALDA,CDA,CLA,LAMA,FRA,SIGA,NPA,UFR,
     &             UAFT,THETDA,NBF,NBA,SPR
C
      COMMON /B/ RRAT,FRDIM,YSIGV,YSIGVA,B701,B702,YBETP,
     &             ASEPA,PI,LAM0,SPR0,PFRONT,PREAR,PCON1,PCON2,
     &             YBETPA,ASEP,N,ITMAX,ERRREL,RPROP
C
      COMMON /SRP/ NDES,RMINC,FRDES,SIGVT,THSRPT,RLV,XRV,THCKV,
     &                 THFDG,ADVR,NBC,DTIP,CLACC,ARAC,AATC,WSS,
     &                 ARAD,GAMC,PO,NHMAX,NMODES,FLEN,
     &                 NPHLAG,NANG,THTDIV,ICALL,TTRT,QTRT,TDUT,
     &                 NPGC,XWL,YWL,ZWL,MTC,FDIA,FRAWC,NTHTDI,
     &                 TDVT,QDUT,QDVT,NROTC
C
      COMMON /PYLON/ APYLVW(51,51),PHPLVW(51,51),APYLCU(51,51),
     &                 PHPLCU(51,51),APYLCV(51,51),PHPLCV(51,51),
     &                 ARBDU
C
      COMMON /RCOM/  RHO,RPM,RPMA,RTIPF,RTIPA,BET701,BET702,
     &                 ASEPIN,MFLT,TO,XPCHA,XPCHAR,NPCLCD,
     &                 NRAD,NRADA,NRADP,THETDP,CDP,IPYLON,
     &                 RADV,CHDV,BETPV1,RADVA,CHDVA,BETPV2,
     &                 RADP,XLEP,CAMPD,XTEP,PSPANV,NROT,
     &                 TMOCV,TMOCV2,YMCV1,YMCV2,ZMCV1,ZMCV2,
     &                 CHD1,CHD2
C
       EXTERNAL FCN
       EXTERNAL FCNP
C
```

```
C*****Read input to program
C          CALL READIN
C
C*****Set program constants
C
           PI   = 3.141592654
           DEGR = PI/180.
           RG   = 1716.2
C*****Parameter vectors for subroutine UMINF: call U4INF to set
C*****values for IPARAM and RPARAM:set values for N,ITMAX,ERRREL:
C*****ERRREL is for NEQNF
           DO 100 I = 1,2
             XSCALE(I) = 1.00
  100      CONTINUE
C
C*****Call IMSL subroutine
C
           CALL U4INF (IPARAM,RPARAM)
           RPARAM(1)  =  10.0E0*RPARAM(1)
               FSCALE = 1.0
                    N = 2
                ITMAX = 200
               ERRREL =.0001
C*****RRAT is ratio of radii of front & rear propellors and is taken
C*****as unity in this study
             RRAT = 1.
C*****RPROP is mean propellor radius in feet & DTIP the corresp. dia.
            RPROP = (RTIPF+RTIPA)/24.
            DTIP  = 2.*RPROP
C*****UFLT= flight speed in fps
             UFLT = SQRT(GAMC*RG*TO)*MFLT
C*****LAM= flight speed/wheel tip speed-front rotor
              LAM = UFLT*30./(RPM*PI*RPROP)
C*****LAMA= flight speed/wheel tip speed-aft rotor
             LAMA = LAM*SPR*RRAT
C*****SPR= speed ratio:rpm(front)/rpm(aft)
              SPR = RPM/RPMA
C*****OM:front rotor angular velocity in radians/second
               OM = RPM*PI/30.
C*****Thrust related constants of conversion
           THCON1 = PI*RHO*OM**2*RPROP**3/(FLOAT(NBF))
           THCON2 = THCON1*FLOAT(NBF)/(SPR**2*FLOAT(NBA))
C*****Power related constants of conversion
            PCON1 = THCON1*FLOAT(NBF)*OM*RPROP/550.
            PCON2 = THCON2*FLOAT(NBA)*(OM/SPR)*RPROP/550.
             LAM0 = LAM
             LAM2 = LAM**2
             SPR0 = SPR
            STERM = SPR*(1.+SPR)
C*****FMT,AMT:wheel tip Mach #'s front & aft
              FMT = MFLT/LAM
              AMT = FMT/SPR
C*****ASEP:pitch change axes separation/propellor radius
C*****ASEPA:likewise for aft rotor
             ASEP = ASEPIN/(RPROP*12.)
            ASEPA = ASEP*RRAT
            THETP = THETDP*DEGR
              NPF = NPCLCD
```

```
              NPA = NPCLCD
C*****Done with program constants
C
C*****Solidity of front rotor:SIGV
        DO 200 I = 1,NRAD
          SIGV(I) = CHDV(I)*FLOAT(NBF)/(2.*PI*RADV(I))
  200   CONTINUE
C*****Span in inches (front)
            DIFFR = RADV(NRAD)-RADV(1)
C*****Solidity of aft rotor:SIGVA
        DO 300 I = 1,NRADA
          SIGVA(I) = CHDVA(I)*FLOAT(NBA)/(2.*PI*RADVA(I))
  300   CONTINUE
C*****Normalise radii of pylon locations by
C*****tip radius of front propellor
        DO 400 I = 1,NRADP
          FRADP(I) = RADP(I)/RADV(NRAD)
  400   CONTINUE
C*****FRDES is used for acoustic calculations and derived from PSPANV
        DO 500 I = 1,NDES
          RADDES(I) = RADV(1)+.01*DIFFR*PSPANV(I)
           FRDES(I) = RADDES(I)/RADV(NRAD)
  500   CONTINUE
C*****Interpolate pylon inputs to FRDES related radial
C*****stations
        CALL LSPFIT (FRADP, XLEP,NRADP,FRDES, XLEPD,NDES,0)
        CALL LSPFIT (FRADP, XTEP,NRADP,FRDES, XTEPD,NDES,0)
        CALL LSPFIT (FRADP,CAMPD,NRADP,FRDES,CAMPDD,NDES,0)
C*****End of interpolation of pylon inputs
C*****AATPR:angle of attack on pylon in radians
            AATPR = AATC*COS(THETP)*DEGR
C*****NK: number of pylon harmonics to be (explicitly) calculated
              NK =NPGC/2+2
         DO 600 I = 1,NRAD
           FRV(I) = RADV(I)/RADV(NRAD)
  600    CONTINUE
C*****RMINC: hub tip ratio
            RMINC =FRV(1)
         DO 700 I = 1,NRADA
           FRVA(I) = RADVA(I)/RADVA(NRADA)
  700    CONTINUE
C*****In DO 800 loop aft locations are "rescaled" so that input data
C*****for aft rotor also goes from FRV(1) to 1 : this is because
C*****we cannot rigorously handle differential diameter effects
C*****(differential diameter refers to front and aft rotors being
C*****of different hub and or tip diameters)
         DO 800 I = 1,NRADA
           FRVAN(I) = (FRVA(I)-FRVA(1))*(1.-FRV(1))/
     &                          (1.-FRVA(1))+FRV(1)
  800    CONTINUE
C*****Begin interpolation of various quantities to FRDES & FRDIM
C*****locations:various quantities are SIGV,SIGVA,BETPV1,BETPV2,
C*****YMCV1,YMCV2,ZMCV1,ZMCV2,TMOCV,TMOCV2,CHD1,CHD2
C*****FRDES is used for acoustic calculations
C*****FRDIM is used for total power & total thrust calculations
         DO 900 I = 1,10
           FRDIM(I) = FRV(1)+(.05+.1*FLOAT(I-1))*(1.-FRV(1))
  900    CONTINUE
```

```
C*****70 % span twists:B701 & B702
            U(1) = FRV(1)+.7*(1.-FRV(1))
       CALL LSPFIT (  FRV,   BETPV1, NRAD,    U,       S,   1,0)
          B701 = S(1)
       CALL LSPFIT (FRVAN,  BETPV2, NRADA,    U,       S,   1,0)
          B702 = S(1)
              MM = 10
       CALL LSPFIT (  FRV,    SIGV, NRAD,FRDIM,   YSIGV,  MM,0)
       CALL LSPFIT (FRVAN,   SIGVA,NRADA,FRDIM,  YSIGVA,  MM,0)
       CALL LSPFIT (  FRV,  BETPV1, NRAD,FRDIM,   YBETP,  MM,0)
       CALL LSPFIT (FRVAN,  BETPV2,NRADA,FRDIM,  YBETPA,  MM,0)
       CALL LSPFIT (  FRV,    SIGV, NRAD,FRDES,   YSGVD,NDES,0)
       CALL LSPFIT (FRVAN,   SIGVA,NRADA,FRDES,  YSGVDA,NDES,0)
       CALL LSPFIT (  FRV,  BETPV1, NRAD,FRDES,    YBPD,NDES,0)
       CALL LSPFIT (FRVAN,  BETPV2,NRADA,FRDES,   YBPDA,NDES,0)
       CALL LSPFIT (  FRV,   YMCV1, NRAD,FRDES,   YMCD1,NDES,0)
       CALL LSPFIT (FRVAN,   YMCV2,NRADA,FRDES,   YMCD2,NDES,0)
       CALL LSPFIT (  FRV,   ZMCV1, NRAD,FRDES,   ZMCD1,NDES,0)
       CALL LSPFIT (FRVAN,   ZMCV2,NRADA,FRDES,   ZMCD2,NDES,0)
       CALL LSPFIT (  FRV,   TMOCV, NRAD,FRDES,  THKVIF,NDES,0)
       CALL LSPFIT (FRVAN,  TMOCV2,NRADA,FRDES,  THKVIR,NDES,0)
       CALL LSPFIT (  FRV,    CHD1, NRAD,FRDES,   CHDD1,NDES,0)
       CALL LSPFIT (FRVAN,    CHD2,NRADA,FRDES,   CHDD2,NDES,0)
C*****End of interpolations
C
C*****Initial guesses (XG1 vector) for call to UMINF
           XG1(1) = BET701
           XG1(2) = BET702
C*****This call "re-adjusts" front and aft rotor 70 % span pitches
C*****to match the input estimates of the power absorbed by the
C*****front and aft rotors
C
C*****Call IMSL subroutine
C
       CALL UMINF (FCNP,N,XG1,XSCALE,FSCALE,IPARAM,RPARAM,XV,
     &             FVALUE)
          BET701 = XV(1)
          BET702 = XV(2)
          DELBP1 = BET701-B701
          DELBP2 = BET702-B702
C
C*****DO 1000,1100 & 1200 loops compute power,thrust of front & aft
C*****rotors with "new" twists by integration over FRDIM stations
C
       DO 1000 IRAD = 1,10
              FR = FRDIM(IRAD)
             FRA = FR*RRAT
             SIG = YSIGV(IRAD)
            SIGA = YSIGVA(IRAD)
          THETDF = BET701-B701+YBETP(IRAD)
          THETDA = BET702-B702+YBETPA(IRAD)
              AS = ASEP/FR
C*****UFR & UAFT are slipstream contraction factors for front &
C*****aft rotors at non dimensional radius FR
             UFR = 1.+AS/SQRT(AS**2+1.)
              AS = ASEPA/FRA
            UAFT = 1.-AS/SQRT(AS**2+1.)
C*****Initial guesses for angles of attack on front & aft rotors
```

```
        XGUESS(1) = 2.
        XGUESS(2) = 2.
              LAM = LAM0
              SPR = SPR0
             LAMA = LAM*SPR*RRAT
C*****Calculate angles of attack by call to NEQNF/FCN
C
C*****Call IMSL subroutine
C
             CALL NEQNF (FCN,ERRREL,N,ITMAX,XGUESS,X,FNORM)
            PHID1 = THETDF-X(1)
            PHID2 = THETDA-X(2)
            PHIR1 = PHID1*DEGR
            PHIR2 = PHID2*DEGR
             SPH1 = SIN(PHIR1)
             CPH1 = COS(PHIR1)
             TPH1 = SPH1/CPH1
          SCPH21 = 1./(CPH1**2)
             SPH2 = SIN(PHIR2)
             CPH2 = COS(PHIR2)
             TPH2 = SPH2/CPH2
          SCPH22 = 1./(CPH2**2)
C*****For angles of attack X(1),X(2) (front & aft rotors),CLCDEV
C*****evaluates lift & drag coefficients for front & aft rotors
             CALL CLCDEV (X,CLA1,CDA1,CLA2,CDA2)
              CX1 = CLA1*SPH1+CDA1*CPH1
              CY1 = CLA1*CPH1-CDA1*SPH1
              CX2 = CLA2*SPH2+CDA2*CPH2
              CY2 = CLA2*CPH2-CDA2*SPH2
            ASPH1 = ABS(SPH1)
C*****PTLF & PTLFA are Prandtl tip loss factors for front and aft
C*****rotors at non dimensional radius FR
              FLC = .5*FLOAT(NBF)*(1.-FR)/(FR*ASPH1)
           ARGACS = EXP(-FLC)
             PTLF = (2./3.1416)*ACOS(ARGACS)
            ASPH2 = ABS(SPH2)
              FLC = .5*FLOAT(NBA)*(1.-FRA)/(FRA*ASPH2)
           ARGACS = EXP(-FLC)
            PTLFA = (2./3.1416)*ACOS(ARGACS)
               Q3 = SIG*CX1/(4.*SPH1*CPH1*PTLF)
               Q4 = SIGA*CX2/(4.*SPH2*CPH2*PTLFA)
               Q1 = (Q3*CY1/CX1)*FR/LAM
               Q2 = (Q4*CY2/CX2)*FRA/LAMA
              AP1 = Q3/(1.+Q3)
              AP2 = (1.+2.*AP1*SPR*PTLF)*Q4/(1.+Q4)
              AA1 = Q1*(1.-AP1)
              AA2 = Q2*(1.+2.*AP1*PTLF*SPR-AP2)
              TF1 = SIG*FR**3*(1.-AP1)**2*CY1*SCPH21
              TF2 = SIGA*FRA**3*(1.+2.*AP1*SPR*PTLF-AP2)**2*
     &                                            CY2*SCPH22
              QF1 = TF1*FR*CX1/CY1
              QF2 = TF2*FRA*CX2/CY2
C*****Thrusts/unit span per blade front & aft
            THRFR = THCON1*TF1
            THRRR = THCON2*TF2
      THRFRV(IRAD)= THRFR
      THRRRV(IRAD)= THRRR
              LAM = LAM0
```

```
             SPR = SPR0
            LAMA = LAM*SPR*RRAT
C*****Power per unit span front & aft rotors (summed over all blades)
      POWF(IRAD) = PCON1*QF1
      POWA(IRAD) = PCON2*QF2
 1000   CONTINUE
C
C*****Power calculation for front & aft rotors
            SUMF = 0.00
            SUMA = 0.00
        DO 1100 I = 1,10
            SUMF = SUMF+POWF(I)
            SUMA = SUMA+POWA(I)
 1100   CONTINUE
            PWRF = SUMF*RPROP*(FRDIM(2)-FRDIM(1))
            PWRA = SUMA*RPROP*(FRDIM(2)-FRDIM(1))
C*****Thrust calculation for front & aft rotors
            SUM1 = 0.00
            SUM2 = 0.00
        DO 1200 I=1,10
            SUM1 = SUM1+THRFRV(I)
            SUM2 = SUM2+THRRRV(I)
 1200   CONTINUE
          TOTTHF = SUM1*RPROP*(FRDIM(2)-FRDIM(1))*FLOAT(NBF)
          TOTTHR = SUM2*RPROP*(FRDIM(2)-FRDIM(1))*FLOAT(NBA)
C
C*****Write out some overall calculated steady state parameters
C
        WRITE(14,5000)
C
        WRITE(14,*)' COMPUTED 70 % SPAN PITCH (deg.): FRONT = ',
     &                              XV(1)
        WRITE(14,*)'                                   REAR = ',
     &                              XV(2)
        WRITE(14,*)'  '
C
        WRITE(14,*)'            TOTAL THRUSTS IN LBF: FRONT = ',
     &                             TOTTHF
        WRITE(14,*)'                                   REAR = ',
     &                             TOTTHR
        WRITE(14,*)'  '
C
            TOTTH = TOTTHF+TOTTHR
C
        WRITE(14,*)'                 TOTAL THRUST IN LBF = ',TOTTH
        WRITE(14,*)'  '
C
        WRITE(14,*)'                  POWER (HP) : FRONT = ',
     &                               PWRF
        WRITE(14,*)'                                   REAR = ',
     &                               PWRA
        WRITE(14,*)'  '
C
             PTOT = PWRF+PWRA
C
        WRITE(14,*)'                    TOTAL POWER (HP) = ',PTOT
C
C*****End write
```

```
C
C*****In DO 1300 loop, the non dimensional axial sweep (XRVV) &
C*****tangential lean (BSWPFD & BSWPAD) are calculated at non dimensional
C*****FRDES radial locations for input to acoustic calculations
C
******************************************************************
C
        DO 1300 IRAD = 1,NDES
            RMCD1 = SQRT(YMCD1(IRAD)*YMCD1(IRAD)+ZMCD1(IRAD)*ZMCD1(IRAD))
            IF(RMCD1.GT..00001) THEN
                PSI1 = ATAN2(ZMCD1(IRAD),YMCD1(IRAD))
                YMCD1P = RMCD1*COS(PSI1+DELBP1*DEGR)
                ZMCD1P = RMCD1*SIN(PSI1+DELBP1*DEGR)
            ELSE
                YMCD1P = 0.
                ZMCD1P = 0.
            END IF
            RMCD2 = SQRT(YMCD2(IRAD)*YMCD2(IRAD)+ZMCD2(IRAD)*ZMCD2(IRAD))
            IF(RMCD2.GT..00001) THEN
                PSI2 = ATAN2(ZMCD2(IRAD),YMCD2(IRAD))
                YMCD2P = RMCD2*COS(PSI2+DELBP2*DEGR)
                ZMCD2P = RMCD2*SIN(PSI2+DELBP2*DEGR)
            ELSE
                YMCD2P = 0.
                ZMCD2P = 0.
            END IF
            YQCD1P = YMCD1P-0.25*CHDD1(IRAD)*COS((YBPD(IRAD)+DELBP1)*DEGR)
            ZQCD1P = ZMCD1P-0.25*CHDD1(IRAD)*SIN((YBPD(IRAD)+DELBP1)*DEGR)
            YQCD2P = YMCD2P-0.25*CHDD2(IRAD)*COS((YBPD(IRAD)+DELBP2)*DEGR)
            ZQCD2P = ZMCD2P-0.25*CHDD2(IRAD)*SIN((YBPD(IRAD)+DELBP2)*DEGR)
C
C *** CHANGE FROM 'ZMC' AND 'YMC' TO 'AXIAL SWEEP' AND
C                     'RADIAL LEAN'
C
            XRVV(1,IRAD) = 2.*ZQCD1P
            BSWPFD(IRAD) = (2.*YQCD1P/FRDES(IRAD))/DEGR
            XRVV(2,IRAD) = 2.*ZQCD2P
            BSWPAD(IRAD) = (2.*YQCD2P/FRDES(IRAD))/DEGR
 1300   CONTINUE
******************************************************************
C
C*****DO 1800 (including DO 1600 loop) calculates partial derivatives
C*****w.r.t LAM & SPR and hence unsteady force coefficients in the
C*****quasi-steady approximation:in DO 1800 loop, axial & tangential
C*****velocity distortion harmonics due to pylon viscous wake / pylon
C*****circulation are also calculated
C
        DO 1800 IRAD = 1,NDES
              FR = FRDES(IRAD)
             FRA = FR*RRAT
             SIG = YSGVD(IRAD)
            SIGA = YSGVDA(IRAD)
          THETDF = BET701-B701+YBPD(IRAD)
          THETDA = BET702-B702+YBPDA(IRAD)
C*****Pitch angles for transmittal to acoustic subroutine SRPIE
      THSRP(1,IRAD)= THETDF
      THSRP(2,IRAD)= THETDA
          CAXFD2 = PI*SIG*SIN(THETDF*DEGR)/FLOAT(NBF)
```

```
C*****Pylon mid chord
            XMIDC = .5*(XLEPD(IRAD)+XTEPD(IRAD))
           RADIUS = FR*12.*RPROP
C*****Axial Projection terms
            PROJT = SIN(YBPD(IRAD)*DEGR)
           PROJTN = SIN(THETDF*DEGR)
C*****Front rotor chord length
             CHRD = 2.*PI*RADIUS*YSGVD(IRAD)/FLOAT(NBF)
C*****Axial location of front rotor quarter chord
              XQC = XPCHA+XRVV(1,IRAD)*RADIUS
C*****Axial location of aft rotor quarter chord
             XQCR = XPCHAR+XRVV(2,IRAD)*RADIUS
C*****Axial location of front rotor leading edge
             XDLE = XQC-0.25*CHRD*PROJTN
C*****Non dimensional distances from front & aft rotor quarter
C*****chord locations to pylon mid chord
            XPYLF = ABS(XMIDC-XQC)/RADIUS
            XPYLA = ABS(XMIDC-XQCR)/RADIUS
C*****Slipstream contraction factors at pylon mid chord
            UPYLF = 1.-XPYLF/SQRT(1.+XPYLF**2)
            UPYLA = 1.-XPYLA/SQRT(1.+XPYLA**2)
C*****Term related to circulation around pylon
C*****Equals zero if IPYLON=0
           CIRCPT = .5*ABS(XTEPD(IRAD)-XLEPD(IRAD))*
     &                (AATPR+.25*CAMPDD(IRAD)*DEGR)/RADIUS
           IF (IPYLON.EQ.0) CIRCPT = 0.00
C*****UFRU & UAFTSP are slipstream contraction factors at front rotor
C*****leading edge
             UFRU = 1.-CAXFD2/SQRT(CAXFD2**2+1.)
               AS = ASEP/FR+CAXFD2
           UAFTSP = 1.-AS/SQRT(AS**2+1.)
C*****UFR & UAFT are slipstream contraction factors for front &
C*****aft rotors at non dimensional radius FR
               AS = ASEP/FR
              UFR = 1.+AS/SQRT(AS**2+1.)
               AS = ASEPA/FRA
             UAFT = 1.-AS/SQRT(AS**2+1.)
C*****Initial guesses for angles of attack on front & aft rotors
             X(1) = 2.
             X(2) = 2.
              LAM = LAM0
              SPR = SPR0
             LAMA = LAM*SPR*RRAT
C*****In DO 1600 loop we vary LAM & SPR by .01 up & down to evaluate
C*****partial derivatives w.r.t LAM & SPR
C
        DO 1600 IJK = 1,5
              LAM = LAM0
              SPR = SPR0
              IF(IJK.EQ.2) LAM = LAM0-.01
              IF(IJK.EQ.3) LAM = LAM0+.01
              IF(IJK.EQ.4) SPR = SPR0-.01
              IF(IJK.EQ.5) SPR = SPR0+.01
             LAMA = LAM*SPR*RRAT
        XGUESS(1) = X(1)
        XGUESS(2) = X(2)
C*****Calculate angles of attack by call to NEQNF/FCN
C
```

```
C*****Call IMSL subroutine
C
            CALL NEQNF (FCN,ERRREL,N,ITMAX,XGUESS,X,FNORM)
            PHID1 = THETDF-X(1)
            PHID2 = THETDA-X(2)
            PHIR1 = PHID1*DEGR
            PHIR2 = PHID2*DEGR
             SPH1 = SIN(PHIR1)
            ASPH1 = ABS(SPH1)
             CPH1 = COS(PHIR1)
             TPH1 = SPH1/CPH1
          SCPH21 = 1./(CPH1**2)
             SPH2 = SIN(PHIR2)
            ASPH2 = ABS(SPH2)
             CPH2 = COS(PHIR2)
             TPH2 = SPH2/CPH2
          SCPH22 = 1./(CPH2**2)
          CALL CLCDEV (X,CLA1,CDA1,CLA2,CDA2)
              CX1 = CLA1*SPH1+CDA1*CPH1
              CY1 = CLA1*CPH1-CDA1*SPH1
              CX2 = CLA2*SPH2+CDA2*CPH2
              CY2 = CLA2*CPH2-CDA2*SPH2
C*****PTLF & PTLFA are Prandtl tip loss factors for front and aft
C*****rotors at non dimensional radius FR
              FLC = .5*FLOAT(NBF)*(1.-FR)/(FR*ASPH1)
           ARGACS = EXP(-FLC)
             PTLF = (2./3.1416)*ACOS(ARGACS)
              FLC = .5*FLOAT(NBA)*(1.-FRA)/(FRA*ASPH2)
           ARGACS = EXP(-FLC)
            PTLFA = (2./3.1416)*ACOS(ARGACS)
               Q3 = SIG*CX1/(4.*SPH1*CPH1*PTLF)
               Q4 = SIGA*CX2/(4.*SPH2*CPH2*PTLFA)
               Q1 = (Q3*CY1/CX1)*FR/LAM
               Q2 = (Q4*CY2/CX2)*FRA/LAMA
              AP1 = Q3/(1.+Q3)
              AP2 = (1.+2.*AP1*SPR*PTLF)*Q4/(1.+Q4)
              AA1 = Q1*(1.-AP1)
              AA2 = Q2*(1.+2.*AP1*PTLF*SPR-AP2)
              TF1 = SIG*FR**3*(1.-AP1)**2*CY1*SCPH21
              TF2 = SIGA*FRA**3*(1.+2.*AP1*SPR*PTLF-AP2)**2*
     &                                      CY2*SCPH22
              QF1 = TF1*FR*CX1/CY1
              QF2 = TF2*FRA*CX2/CY2
C*****FUFV1:non dimensional velocity at front rotor l. edge
       FUFV1(IJK) = 1.+AA1*PTLF*UFRU+PTLFA*UAFTSP*AA2
        TF1V(IJK) = TF1
        TF2V(IJK) = TF2
        QF1V(IJK) = QF1
        QF2V(IJK) = QF2
C*****Skip calculations from here to "1500 CONTINUE" if IJK.GT.1
           IF (IJK.GT.1) GO TO 1500
C*****UMIDP:Non dimensional velocity at pylon mid chord
           UMIDP = 1.+AA1*PTLF*UPYLF+AA2*PTLFA*UPYLA
C*****PYLHC:pylon half chord
           PYLHC = .5*ABS(XTEPD(IRAD)-XLEPD(IRAD))
C*****XOVCKS:distance from pylon mid chord to front rotor
C*****leading edge/PYLHC
          XOVCKS = ABS(XDLE-XMIDC)/PYLHC
```

```
C*****XOVA:distance from pylon mid chord to front rotor
C*****leading edge/RADIUS
             XOVA = XOVCKS*PYLHC/RADIUS
C*****UCOVKS:velocity defect at front rotor l.edge (non dimensional)
C*****due to viscous wake of pylon
C*****Equals zero if IPYLON=0
           UCOVKS = -UMIDP*2.42*SQRT(CDP)/(XOVCKS-.4)
           IF (IPYLON.EQ.0) UCOVKS = 0.00
C*****Angular half width of pylon viscous wake
              YKS = .68*SQRT(2.*CDP)*PYLHC*SQRT(XOVCKS-.7)/RADIUS
C*****In DO 1400 loop explicit evaluation of axial & tangential velocity
C*****distortion harmonics (amplitude & phase) due to viscous wake
C*****& circulation of pylon are carried out
        DO 1400 INK = 1,NK
                     FI = FLOAT(INK)
                     T1 = SIN(FI*YKS)/PI
                     T2 = 1./FI+YKS**2*FI/(PI**2-(YKS*FI)**2)
       APYLVW(INK,IRAD) = .5*T1*UCOVKS*T2
       APYLCU(INK,IRAD) = -.5*CIRCPT*UMIDP*EXP(-FI*XOVA)
       APYLCV(INK,IRAD) = APYLCU(INK,IRAD)
      PHPLVW(INK,IRAD) = THETP
      PHPLCU(INK,IRAD) = THETP-.5*PI/FI
      PHPLCV(INK,IRAD) = THETP
 1400     CONTINUE
C*****Skip to this line if IJK.GT.1
 1500   CONTINUE
              LAM = LAM0
              SPR = SPR0
 1600   CONTINUE
C*****End of DO 1600 loop
C
             LAMA = LAM*SPR*RRAT
C*****Steady Thrust Terms
              TF1 = TF1V(1)
              TF2 = TF2V(1)
      TTR(1,IRAD) = TF1
      TTR(2,IRAD) = TF2
C*****Steady Tangential Force & Torque Terms
              QF1 = QF1V(1)
              QF2 = QF2V(1)
      QTR(1,IRAD) = QF1/FR
      QTR(2,IRAD) = QF2/FR
C*****FUF1:non dimensional velocity at front rotor l. edge
             FUF1 = FUFV1(1)
C*****Central difference based partial derivatives w.r.t LAM & SPR
            FUFL1 = 50.*(FUFV1(3)-FUFV1(2))
            FUFS1 = 50.*(FUFV1(5)-FUFV1(4))
             TF1L = 50.*(TF1V(3)-TF1V(2))
             TF1S = 50.*(TF1V(5)-TF1V(4))
             QF1L = 50.*(QF1V(3)-QF1V(2))
             QF1S = 50.*(QF1V(5)-QF1V(4))
             TF2L = 50.*(TF2V(3)-TF2V(2))
             TF2S = 50.*(TF2V(5)-TF2V(4))
             QF2L = 50.*(QF2V(3)-QF2V(2))
             QF2S = 50.*(QF2V(5)-QF2V(4))
C*****Partial derivatives of axial velocity at front rotor leading
C*****edge w.r.t (u,v) at upstream infinity
             UFU1 = FUF1+FUFL1*LAM
```

```
              UFV1 = LAM2*FUFL1/FR-STERM*FUFS1*LAM/FR
C*****Non dimensional force derivatives w.r.t (u/U) & (v/U)
             FRT1U = LAM*TF1L
             FRT1V = (-2.*LAM*TF1/FR+LAM2*TF1L/FR-STERM*LAM*TF1S/FR)
             FRQ1U = LAM*QF1L/FR
             FRQ1V = (-2.*LAM*QF1/FR+LAM2*QF1L/FR-STERM*LAM*QF1S/FR)/
     &                                                             FR
             FRT2U = LAM*TF2L
             FRT2V = (2.*LAM*SPR*TF2/FR+LAM2*TF2L/FR-STERM*LAM*
     &                                                    TF2S/FR)
             FRQ2U = LAM*QF2L/FR
             FRQ2V = (2.*LAM*SPR*QF2/FR+LAM2*QF2L/FR-STERM*LAM*
     &                                                 QF2S/FR)/FR
             TDUTR(1,IRAD) = FRT1U
             TDUTR(2,IRAD) = FRT2U
             TDVTR(1,IRAD) = FRT1V
             TDVTR(2,IRAD) = FRT2V
             QDUTR(1,IRAD) = FRQ1U
             QDUTR(2,IRAD) = FRQ2U
             QDVTR(1,IRAD) = FRQ1V
             QDVTR(2,IRAD) = FRQ2V
C*****In DO 1700 loop we transfer pylon harmonics to upstream infinity
         DO 1700 INK = 1,NK
             APYLVW(INK,IRAD) = APYLVW(INK,IRAD)/UFU1
             APYLCU(INK,IRAD) =(APYLCU(INK,IRAD)-UFV1*
     &                          APYLCV(INK,IRAD))/UFU1
 1700      CONTINUE
 1800    CONTINUE
C
C*****End of DO 1900 loop
C*****In DO 2000 loop,we set up inputs for front and aft rotors for
C*****and also call  SRPIE (acoustic program)
         DO 2000 IJK = 1,2
             ICALL = IJK
             IF(ICALL.EQ.1)NROTC = NROT
             IF(ICALL.EQ.2)NROTC = -NROT
             IF(ICALL.EQ.1)  NBC = NBF
             IF(ICALL.EQ.2)  NBC = NBA
             IF(ICALL.EQ.1)  MTC = FMT
             IF(ICALL.EQ.2)  MTC = AMT
             IF(ICALL.EQ.1) ADVR = LAM*PI
             IF(ICALL.EQ.2) ADVR = LAMA*PI
           DO 1900 IRAD=1,NDES
               IF(ICALL.EQ.1) SIGVT(IRAD) = YSGVD(IRAD)
               IF(ICALL.EQ.2) SIGVT(IRAD) = YSGVDA(IRAD)
                             THSRPT(IRAD) = THSRP(IJK,IRAD)
               IF(ICALL.EQ.1)    RLV(IRAD) = -BSWPFD(IRAD)*FLOAT(NROT)
               IF(ICALL.EQ.2)    RLV(IRAD) = BSWPAD(IRAD)*FLOAT(NROT)
                                 XRV(IRAD) = XRVV(IJK,IRAD)
               IF(ICALL.EQ.1) THCKV(IRAD) = THKVIF(IRAD)
               IF(ICALL.EQ.2) THCKV(IRAD) = THKVIR(IRAD)
                               TTRT(IRAD) = TTR(IJK,IRAD)
                               QTRT(IRAD) = QTR(IJK,IRAD)
                               TDUT(IRAD) = TDUTR(IJK,IRAD)
                               TDVT(IRAD) = TDVTR(IJK,IRAD)
                               QDUT(IRAD) = QDUTR(IJK,IRAD)
                               QDVT(IRAD) = QDVTR(IJK,IRAD)
 1900      CONTINUE
```

```
C
                    CALL SRPIE(ISOLAT)
C
 2000   CONTINUE
C
      RETURN
C
C --- FORMAT STATEMENTS ---------------------------------------------
C
 5000 FORMAT('1',//20X,' SOME OVERALL CALCULATED STEADY STATE',
     &                   ' QUANTITIES',///)
C
      END
C
C
C          * * * * E N D   O F   M A I N   P R O G R A M * * * *
C
C
C *******************************
C *      SUBROUTINE FCN         *
C ********************************************************************
C
C     SUBROUTINE FCN IS USED TO SOLVE FOR THE ANGLES OF ATTACK
C     [ X(1) AND X(2) ] ON THE FRONT AND REAR ROTORS IN THE STEADY
C     STATE AT NON DIMENSIONAL RADIUS FR GIVEN THE PITCH ANGLES THETDF/
C     THETDA, NUMBER OF BLADES NBF/NBA, ADVANCE RATIOS LAM/LAMA, SPEED RATIO
C     SPR, SOLIDITIES SIG/SIGA AND SLIPSTREAM CONTRACTION FACTORS UFR/UAFT .
C     IT CALLS THE LIFT/DRAG COEFFICIENT SUBROUTINE CLCDEV . THE ANGLES
C     OF ATTACK [ X(1) AND X(2) ] MUST BE SUCH THAT THE TWO QUANTITIES
C     F(1) AND F(2) DEFINED IN THIS SUBROUTINE ARE BOTH ZERO.
C
C ********************************************************************
C
      SUBROUTINE FCN(X,F,N)
C
      DIMENSION ALD(51),CD(51),CL(51)
C
      DIMENSION ALDA(51),CDA(51),CLA(51)
C
      DIMENSION X(3),F(3)
C
      REAL      LAM,LAMA
C
      COMMON /A/ ALD,CD,CL,LAM,FR,SIG,THETDF,DEGR,NPF,
     &           ALDA,CDA,CLA,LAMA,FRA,SIGA,NPA,UFR,
     &           UAFT,THETDA,NBF,NBA,SPR
C
            PHID1 = THETDF-X(1)
            PHID2 = THETDA-X(2)
            PHIR1 = PHID1*DEGR
            PHIR2 = PHID2*DEGR
             SPH1 = SIN(PHIR1)
            ASPH1 = ABS(SPH1)
             CPH1 = COS(PHIR1)
             TPH1 = SPH1/CPH1
          SCPH21 = 1./(CPH1**2)
             SPH2 = SIN(PHIR2)
            ASPH2 = ABS(SPH2)
```

```
            CPH2 = COS(PHIR2)
            TPH2 = SPH2/CPH2
         SCPH22 = 1./(CPH2**2)
         CALL CLCDEV (X,CLA1,CDA1,CLA2,CDA2)
             CX1 = CLA1*SPH1+CDA1*CPH1
             CY1 = CLA1*CPH1-CDA1*SPH1
             CX2 = CLA2*SPH2+CDA2*CPH2
             CY2 = CLA2*CPH2-CDA2*SPH2
C*****PTLF & PTLFA are Prandtl tip loss factors for front and aft
C*****rotors at non dimensional radius FR
             FLC = .5*FLOAT(NBF)*(1.-FR)/(FR*ASPH1)
          ARGACS = EXP(-FLC)
            PTLF = (2./3.1416)*ACOS(ARGACS)
             FLC = .5*FLOAT(NBA)*(1.-FRA)/(FRA*ASPH2)
          ARGACS = EXP(-FLC)
           PTLFA = (2./3.1416)*ACOS(ARGACS)
              Q3 = SIG*CX1/(4.*SPH1*CPH1*PTLF)
              Q4 = SIGA*CX2/(4.*SPH2*CPH2*PTLFA)
              Q1 = (Q3*CY1/CX1)*FR/LAM
              Q2 = (Q4*CY2/CX2)*FRA/LAMA
             AP1 = Q3/(1.+Q3)
             AP2 = (1.+2.*AP1*SPR*PTLF)*Q4/(1.+Q4)
             AA1 = Q1*(1.-AP1)
             AA2 = Q2*(1.+2.*AP1*PTLF*SPR-AP2)
             DEN = (1.+AA1+AA2*PTLFA*UAFT)
            F(1) = LAM*CPH1-FR*(1.-AP1)*SPH1/DEN
            DENA = (1.+AA1*UFR*PTLF+AA2)
            F(2) = LAMA*CPH2-FRA*(1.+2.*AP1*PTLF*SPR-AP2)*SPH2/DENA
        RETURN
      END
C ******************************
C *      SUBROUTINE CLCDEV      *
C *****************************************************************
C
C     SUBROUTINE CLCDEV EVALUATES THE LIFT AND DRAG COEFFICIENTS
C     ON THE FRONT (CLP AND CDP) AND REAR ROTORS (CLPA AND CDPA)
C     GIVEN THE CORRESPONDING ANGLES OF ATTACK X(1) AND X(2). IT CALLS
C     THE SUBROUTINE LSPFIT AND USES INPUT DATA IN ALD/CL/CD AND
C     ALDA/CLA/CDA.
C
C *****************************************************************
C
      SUBROUTINE CLCDEV(X,CLP,CDP,CLPA,CDPA)
C
      DIMENSION ALD(51),CD(51),CL(51)
C
      DIMENSION ALDA(51),CDA(51),CLA(51)
C
      DIMENSION U(1),S(1),X(3)
C
      REAL      LAM,LAMA
C
      COMMON /A/ ALD,CD,CL,LAM,FR,SIG,THETDF,DEGR,NPF,
     &           ALDA,CDA,CLA,LAMA,FRA,SIGA,NPA,UFR,
     &           UAFT,THETDA,NBF,NBA,SPR
              U(1) = X(1)
           CALL LSPFIT ( ALD, CL, NPF,U,S,1,0)
              CLP = S(1)
```

```
            CALL LSPFIT ( ALD, CD, NPF,U,S,1,0)
               CDP = S(1)
              U(1) = X(2)
            CALL LSPFIT (ALDA,CLA,NPA,U,S,1,0)
              CLPA = S(1)
            CALL LSPFIT (ALDA,CDA,NPA,U,S,1,0)
              CDPA = S(1)
        RETURN
      END
C ******************************
C *      SUBROUTINE FCNP        *
C **********************************************************************
C
C     SUBROUTINE FCNP IS USED TO SOLVE FOR THE 70 % SPAN PITCH ANGLES
C     [ XV(1) AND XV(2) ] ON THE FRONT AND REAR ROTORS SUCH THAT THE
C     POWERS ABSORBED BY THESE ROTORS APPROXIMATELY EQUAL THE INPUT
C     VALUES PFRONT AND PREAR. IN THE DO 100 LOOP THE POWER PER UNIT
C     SPAN IS CALCULATED ON EACH ROTOR. THESE ARE SUMMED IN DO 200 LOOP
C     TO YIELD THE POWER ABSORBED BY EACH ROTOR.XV(1) AND XV(2) ARE
C     TO BE DETERMINED SUCH THE NORM OF THE VECTOR FNEW DEFINED BY THIS
C     SUBROUTINE IS MINIMISED. IT CALLS IMSL SUBROUTINE NEQNF AND THE
C     SUBROUTINE CLCDEV .
C
C **********************************************************************
C
      SUBROUTINE FCNP(NNEW,XV,F)
C
      DIMENSION ALD(51),CD(51),CL(51)
C
      DIMENSION ALDA(51),CDA(51),CLA(51)
C
      DIMENSION RADV(51),RADVA(51),CHDV(51),CHDVA(51),BETPV1(51),
     &          BETPV2(51),SIGV(51),SIGVA(51),
     &          YSIGV(51),FRV(51),FRVA(51),FRVAN(51),
     &          YSIGVA(51),U(1),S(1),POWF(51),POWA(51),
     &          YBETPA(51),FRDIM(51),YBETP(51)
C
      DIMENSION XV(2),FNEW(2),X(2),XGUESS(2)
C
      REAL      LAM,LAMA,LAM0,LAM2
C
      COMMON /A/ ALD,CD,CL,LAM,FR,SIG,THETDF,DEGR,NPF,
     &           ALDA,CDA,CLA,LAMA,FRA,SIGA,NPA,UFR,
     &           UAFT,THETDA,NBF,NBA,SPR
      COMMON /B/ RRAT,FRDIM,YSIGV,YSIGVA,B701,B702,YBETP,
     &           ASEPA,PI,LAM0,SPR0,PFRONT,PREAR,PCON1,PCON2,
     &           YBETPA,ASEP,N,ITMAX,ERRREL,RPROP
C
      EXTERNAL FCN
C
        DO 100 IRAD = 1,10
               FR = FRDIM(IRAD)
              FRA = FR*RRAT
              SIG = YSIGV(IRAD)
             SIGA = YSIGVA(IRAD)
           THETDF = XV(1)-B701+YBETP(IRAD)
           THETDA = XV(2)-B702+YBETPA(IRAD)
               AS = ASEP/FR
```


C-2

```
C*****UFR & UAFT are slipstream contraction factors for front &
C*****aft rotors at non dimensional radius FR
              UFR = 1.+AS/SQRT(AS**2+1.)
               AS = ASEPA/FRA
             UAFT = 1.-AS/SQRT(AS**2+1.)
C*****Initial guesses for angles of attack on front & aft rotors
        XGUESS(1) = 2.
        XGUESS(2) = 2.
              LAM = LAM0
              SPR = SPR0
             LAMA = LAM*SPR*RRAT
C*****Calculate angles of attack by call to NEQNF/FCN
C
C*****Call IMSL subroutine
C
            CALL NEQNF (FCN,ERRREL,N,ITMAX,XGUESS,X,FNORM)
            PHID1 = THETDF-X(1)
            PHID2 = THETDA-X(2)
            PHIR1 = PHID1*DEGR
            PHIR2 = PHID2*DEGR
             SPH1 = SIN(PHIR1)
             CPH1 = COS(PHIR1)
             TPH1 = SPH1/CPH1
           SCPH21 = 1./(CPH1**2)
             SPH2 = SIN(PHIR2)
             CPH2 = COS(PHIR2)
             TPH2 = SPH2/CPH2
          SCPH22 = 1./(CPH2**2)
C*****For angles of attack X(1),X(2) (front & aft rotors),CLCDEV
C*****evaluates lift & drag coefficients for front & aft rotors
             CALL CLCDEV (X,CLA1,CDA1,CLA2,CDA2)
              CX1 = CLA1*SPH1+CDA1*CPH1
              CY1 = CLA1*CPH1-CDA1*SPH1
              CX2 = CLA2*SPH2+CDA2*CPH2
              CY2 = CLA2*CPH2-CDA2*SPH2
            ASPH1 = ABS(SPH1)
C*****PTLF & PTLFA are Prandtl tip loss factors for front and aft
C*****rotors at non dimensional radius FR
              FLC = .5*FLOAT(NBF)*(1.-FR)/(FR*ASPH1)
           ARGACS = EXP(-FLC)
             PTLF = (2./3.1416)*ACOS(ARGACS)
            ASPH2 = ABS(SPH2)
              FLC = .5*FLOAT(NBA)*(1.-FRA)/(FRA*ASPH2)
           ARGACS = EXP(-FLC)
            PTLFA = (2./3.1416)*ACOS(ARGACS)
               Q3 = SIG*CX1/(4.*SPH1*CPH1*PTLF)
               Q4 = SIGA*CX2/(4.*SPH2*CPH2*PTLFA)
               Q1 = (Q3*CY1/CX1)*FR/LAM
               Q2 = (Q4*CY2/CX2)*FRA/LAMA
              AP1 = Q3/(1.+Q3)
              AP2 = (1.+2.*AP1*SPR*PTLF)*Q4/(1.+Q4)
              AA1 = Q1*(1.-AP1)
              AA2 = Q2*(1.+2.*AP1*PTLF*SPR-AP2)
              TF1 = SIG*FR**3*(1.-AP1)**2*CY1*SCPH21
              TF2 = SIGA*FRA**3*(1.+2.*AP1*SPR*PTLF-AP2)**2*
     &                                            CY2*SCPH22
              QF1 = TF1*FR*CX1/CY1
              QF2 = TF2*FRA*CX2/CY2
```

```
              LAM = LAM0
              SPR = SPR0
             LAMA = LAM*SPR*RRAT
C*****Power per unit span front & aft rotors (summed over all blades)
       POWF(IRAD) = PCON1*QF1
       POWA(IRAD) = PCON2*QF2
  100   CONTINUE
             SUMF = 0.00
             SUMA = 0.00
        DO 200 I = 1,10
             SUMF = SUMF+POWF(I)
             SUMA = SUMA+POWA(I)
  200   CONTINUE
             PWRF = SUMF*RPROP*(FRDIM(2)-FRDIM(1))
             PWRA = SUMA*RPROP*(FRDIM(2)-FRDIM(1))
          FNEW(1) = 10.*(PWRF-PFRONT)/PFRONT
          FNEW(2) = 10.*(PWRA-PREAR)/PREAR
                F =  FNEW(1)**2+FNEW(2)**2
        RETURN
      END
C ********************************************************************
C *                        ACOUSTIC SUBROUTINE                       *
C *      SINGLE ROTATION PROPELLER INSTALLATION EFFECTS - 'SRPIE'    *
C ********************************************************************
C
C  THIS SUBROUTINE CALCULATES INSTALLATION EFFECTS ON THE FRONT
C  AND REAR ROTORS. INPUTS TO IT ARE TRANSMITTED VIA COMMON /SRP/.
C  IT CALLS SUBROUTINES BJFNA, FCOEFF, OUTPUT AND PHLAG.THE EFFECTS
C  ON EACH ROTOR ARE CALCULATED SEPARATELY.
C
C ********************************************************************
C
C * INPUTS TRANSMITTED TO 'SRPIE' BY CALCULATION VIA COMMON /SRP/
C          FROM MAIN PROGRAM FOR EACH ROTOR
C
C      RMINC  = HUB/TIP RATIO OF PROPELLER
C      FRDES  = NORMALIZED RADIUS OF EACH RADIAL STATION (RAD./TIP RAD.)
C      SIGVT  = SOLIDITY AT EACH RADIAL STATION
C      THSRPT = PITCH ANGLES AT EACH RADIAL STATION (DEGREES)
C      RLV    = RADIAL LEAN AT EACH RADIAL STATION (DEGREES)
C      XRV    = (BLADE AXIAL SWEEP)/(TIP RADIUS) AT EACH RADIAL STATION
C      THCKV  = (MAX THICKNESS)/CHORD AT EACH RADIAL STATION
C      ADVR   = ADVANCE RATIO (J)
C      NBC    = NUMBER OF PROPELLOR BLADES
C      DTIP   = PROPELLER TIP DIAMETER (FT.)
C      ICALL  = 1 denotes FRONT rotor and 2 denotes REAR rotor
C      TTRT   = STEADY THRUST RELATED TERM AT EACH RADIAL STATION
C      QTRT   = STEADY TANGENTIAL FORCE RELATED TERM AT EACH RADIAL STATION
C      TDUT   = TERM RELATED TO QUASI STEADY DERIVATIVE OF THRUST w.r.t.
C                   (u/U) AT EACH RADIAL STATION
C      MTC    = PROPELLER WHEEL TIP MACH NUMBER
C      TDVT   = TERM RELATED TO QUASI STEADY DERIVATIVE OF THRUST w.r.t.
C                   (v/U) AT EACH RADIAL STATION
C      QDUT   = TERM RELATED TO QUASI STEADY DERIVATIVE OF TANGENTIAL FORCE
C                 w.r.t. (u/U) AT EACH RADIAL STATION
C      QDVT   = TERM RELATED TO QUASI STEADY DERIVATIVE OF TANGENTIAL FORCE
C                 w.r.t. (v/U) AT EACH RADIAL STATION
C      NROTC  = FLAG FOR PROPELLER ROTATION DIRECTION (FLA)
```

```
C                  -1 = CLOCKWISE
C                   1 = COUNTER CLOCKWISE
C
C * INPUTS INDIRECTLY TRANSMITTED TO 'SRPIE' BY CALCULATION FROM
C     MAIN PROGRAM: NOTE THAT 'SRPIE' CALLS 'FCOEFF' WHICH SHARES
C     COMMON /PYLON/ WITH MAIN PROGRAM
C
C      APYLVW = AMPLITUDE OF DISTORTION HARMONICS ASSOCIATED WITH
C               PYLON VISCOUS WAKE AT EACH RADIAL STATION:
C               NOTE THAT VISCOUS WAKE DISTORTIONS ARE TREATED
C               AS AXIAL OR "u" VELOCITY DISTORTIONS .
C    PHIPYLVW = PHASE OF DISTORTION HARMONICS ASSOCIATED WITH
C               PYLON VISCOUS WAKE AT EACH RADIAL STATION:
C               NOTE THAT VISCOUS WAKE DISTORTIONS ARE TREATED
C               AS AXIAL OR "u" VELOCITY DISTORTIONS .
C      APYLCU = AMPLITUDE OF DISTORTION HARMONICS ASSOCIATED WITH
C               PYLON CIRCULATION AT EACH RADIAL STATION - "u" OR
C               AXIAL VELOCITY COMPONENT
C    PHIPYLCU = PHASE OF DISTORTION HARMONICS ASSOCIATED WITH
C               PYLON CIRCULATION AT EACH RADIAL STATION - "u" OR
C               AXIAL VELOCITY COMPONENT
C      APYLCV = AMPLITUDE OF DISTORTION HARMONICS ASSOCIATED WITH
C               PYLON CIRCULATION AT EACH RADIAL STATION - "v" OR
C               TANGENTIAL VELOCITY COMPONENT
C    PHIPYLCV = PHASE OF DISTORTION HARMONICS ASSOCIATED WITH
C               PYLON CIRCULATION AT EACH RADIAL STATION - "v" OR
C               TANGENTIAL VELOCITY COMPONENT
C
C **********************************************************************
C
      SUBROUTINE SRPIE(ISOLAT)
C
      REAL    LAM,LAM1,MT,KA,KAC,MP,MM,MFLT,MTC,MY,MYPHI,KACMOD,
     &        MYCON,M0
C
      COMPLEX FZP(10),FZM(10),FTP(10),FTM(10),IM,Z,ZC,ZARG,CSUMZ,
     &        ZP,ZM,RADPH,PCV(5,30,20,51),CSUMP,CZSL,CTSL,
     &        PCVSL(5,20,51),CSUMT,CSMPSL,CCTHK1,CCTHCK,UTHP(10),
     &        UTHM(10),CSUTH,RADPH0,PHMOD
C
      DOUBLE PRECISION KAS,BM(100),BFP(10),BFM(10),BF0,KASY,BF0Y
C
      DIMENSION FRV(51),SIGV(51),THETDV(51),RLV(51),XRV(51),
     &          DIFFR(51),TMINV(10),THCKV(51),TPLV(10),
     &          THTDIV(20,2),PHDV(30),PDBV(5,30,20,2),
     &          PDBSL(5,20,2),DIFFDB(5,30,20,2)
C
      DIMENSION FRDES(51),SIGVT(51),THSRPT(51),TTRT(51),QTRT(51),
     &          TDUT(51),TDVT(51),QDUT(51),QDVT(51)
C
      COMMON /G/     LAM1,CLAC,ARA,AAT,DDP,NPG,XDG,YDG,ZDG,PHU(51),
     &               PHV(51),AMPU(51),AMPV(51)
C
      COMMON /H/     FRAT,FLEN1,SRAT,FRAW,EMFUS,AFUS,IRAD
C
      COMMON /SRP/ NDES,RMINC,FRDES,SIGVT,THSRPT,RLV,XRV,THCKV,
     &             THFDG,ADVR,NBC,DTIP,CLACC,ARAC,AATC,WSS,
     &             ARAD,GAMC,PO,NHMAX,NMODES,FLEN,
```

```
     &              NPHLAG,NANG,THTDIV,ICALL,TTRT,QTRT,TDUT,
     &              NPGC,XWL,YWL,ZWL,MTC,FDIA,FRAWC,NTHTDI,
     &              TDVT,QDUT,QDVT,NROTC
C
        COMMON /CRINST/ BPHM(5,17,2)
C
C
C *** SET INPUT PARAMETERS
C
      PI     = 3.14159265
      CLAC   = CLACC
      ARA    = ARAC
      AAT    = AATC
      NPG    = NPGC
      FRAW   = FRAWC
      LAM1   = ADVR/PI
      NROT   = NROTC
      NRAD   = NDES
      RMIN   = RMINC
      NB     = NBC
      MT     = MTC
      MFLT   = LAM1*MT
      GAM    = GAMC
      DO 5 I = 1,NRAD
          FRV(I)  = FRDES(I)
          SIGV(I) = SIGVT(I)
        THETDV(I) = THSRPT(I)
    5 CONTINUE
C
C *** SET PROGRAM CONSTANTS
C
C  * 'IM' STANDS FOR COMPLEX # i = SQRT(-1.)
C
      IM     = (0.,1.)
      PI     = 3.14159265
      DEGR   = PI/180.
      DDP    = DTIP/WSS
      RODM   = ARAD/DTIP
      FRAT   = FLEN/WSS
      SRAT   = FDIA/FLEN
      XDG    = XWL/WSS
      YDG    = YWL/WSS
      ZDG    = ZWL/WSS
      AC1    = GAM*MT**2/RODM
      CCTHK1 = -IM*GAM*(PI/FLOAT(NB))/RODM
      MY     = MFLT*SIN(AAT*DEGR)
C
C  *  SET UP PHI'S FROM 0 TO 345 IN STEPS OF 15 DEGREES
C
      NPHID  = 24
C
      DO 10 I = 1,NPHID
        PHDV(I) = 15.*FLOAT(I-1)
 10   CONTINUE
C
C  *  SET UP DELTA R'S
C
      DIFFR(1)    = .5*(FRV(2)+FRV(1))-RMIN
```

```
      DIFFR(NRAD) = 1.-.5*(FRV(NRAD)+FRV(NRAD-1))
      NRADM1      = NRAD-1
C
      DO 20 I = 2,NRADM1
        DIFFR(I) = .5*(FRV(I+1)-FRV(I-1))
 20   CONTINUE
C
C ************************************
C *** BEGIN ACOUSTIC CALCULATIONS ***
C ************************************
C
C *** THE DO 150 LOOP EVALUATES ACOUSTIC CONTRIBUTIONS FROM
C *** EACH RADIAL ELEMENT IN LINE SOURCE MODEL
C
      DO 150 IRAD = 1,NRAD
        NP    = NPV
        FR    = FRV(IRAD)
        THCK  = THCKV(IRAD)*THFDG
        SIG   = SIGV(IRAD)
        THETD = THETDV(IRAD)
        THETR = THETD*DEGR
        CTST  = COS(THETR)*SIN(THETR)/FR
        CT2   = -FLOAT(NROT)*COS(THETR)**2/FR
C
        ICOUNT = 0
C
C *** THE DO 140 LOOP IS FOR DIFFERENT HARMONICS
C
        DO 140 IK = 1,NHMAX
          NH    = IK
          NZERO = NB*NH
          NBMAX = NZERO+3+NMODES
          NZT   = NZERO*NROT
          NZTD  = MOD(NZT,2)
C
C *** PHASING DUE TO RADIAL LEAN
C
          RADPH0 = CEXP(-IM*FLOAT(NZT)*DEGR*RLV(IRAD))
          KA     = FR*FLOAT(NZERO)*MT
C
C *** THE DO 130 LOOP IS FOR DIFFERENT THETA'S
C
          DO 130 IJKL = 1,NTHTDI
            THETDI = THTDIV(IJKL,ICALL)
            THETM  = (180.-THETDI)*DEGR
            STHTM0= SIN(THETM)
C
C *** SWITCH FROM CURRENT TO RETARDED THETA'S IF NECESSARY
C ***(SWITCH FROM OBSERVER TO EMISSION ANGLES)
C
            T1 = MFLT*SIN(THETM)**2
            T2 = MFLT*T1
            T3 = COS(THETM)
            T4 = -T1+T3*SQRT(1.-T2)
C
C *** IF OBSERVER ANGLES (NANG = 1) WERE INPUT AS VALUES FOR 'THTDIV',
C *** THEN CALCULATE THE RESPECTIVE EMISSION ANGLES.
C
```

```
            IF(NANG.EQ.1)THETM = ACOS(T4)
            STHETM = SIN(THETM)
            CTHETM = COS(THETM)
            RBOVR  = STHTM0/STHETM
            CAMP   = 1./(1.+MFLT*CTHETM)
            AC     = AC1*CAMP
            CCTHCK = CCTHK1*THCK*(SIG*KA*CAMP)**2*DIFFR(IRAD)*CAMP
            KAS    = KA*SIN(THETM)*CAMP
            KAC    = KA*CTHETM*CAMP
C
C *** ADD PHASING DUE TO SWEEP
C
            RADPH = RADPH0*CEXP(-IM*FLOAT(NZERO)*MT*XRV(IRAD)
     &                *CTHETM*CAMP)
            IERB  = 100
C
C *** COMPUTE BESSEL FUNCTIONS  ***
C
            CALL BJFNA (KAS,NBMAX,BM)
            BF0 = BM(NZERO+1)
            IF((NZT.LT.0).AND.(NZTD.NE.0))BF0 = -BF0
C
C *** THE DO 50 LOOP COMPUTES TERMS NEEDED FOR UNSTEADY THICKNESS NOISE
C *** TPLV & TMINV AND COLLECTS BESSEL FUNCT'S FOR VARIOUS AZIMUTHAL MODES
C
            DO 50 I = 1,NMODES
              TPLV(I)  = (FLOAT(NZERO+I*NROT)/FLOAT(NZERO))
     &                   /CAMP-MFLT*CTHETM
              TMINV(I) = (FLOAT(NZERO-I*NROT)/FLOAT(NZERO))
     &                   /CAMP-MFLT*CTHETM
              NZTP     = NZT+I
              NZTPD    = MOD(NZTP,2)
              NZEROP   = IABS(NZTP)+1
              NZTM     = NZT-I
              NZTMD    = MOD(NZTM,2)
              NZEROM   = IABS(NZTM)+1
              BFP(I)   = BM(NZEROP)
              IF((NZTP.LT.0).AND.(NZTPD.NE.0))BFP(I) = -BFP(I)
              BFM(I)   = BM(NZEROM)
              IF((NZTM.LT.0).AND.(NZTMD.NE.0))BFM(I) = -BFM(I)
 50         CONTINUE
C
            IF(ICOUNT.GT.0)GO TO 100
C
C
C
C *** EVALUATE PARTIAL DERIVATIVES OF AXIAL & TANGENTIAL
C *** FORCES w.r.t. u,v, AND PERFORM SOME NORMALIZATIONS
C
            C1   = DIFFR(IRAD)/(4.*FR)
            TF   = C1*TTRT(IRAD)
            QF   = C1*QTRT(IRAD)
            PDTU = C1*TDUT(IRAD)
            PDTV = C1*TDVT(IRAD)
            PDQU = C1*QDUT(IRAD)
            PDQV = C1*QDVT(IRAD)
C
C *** SUBROUTINE FCOEFF EVALUATES u,v DISTORTIONS AT PLANE OF
```

```
C *** PROPELLER DISK DUE TO ANGLE OF ATTACK, WING+TRAILING VORTEX
C *** SYSTEM,PYLON AND FUSELAGE AND ALSO CARRIES OUT A FOURIER ANALYSIS
C *** OF THESE DISTORTIONS IN THE PHI DIRECTION.
C
            CALL FCOEFF (FR)
C
C *** WE NOW CALCULATE (TILL 90) THE UNSTEADY FORCE HARMONICS
C *** (AXIAL&TANGENTIAL) AND THE UNSTEADY THICKNESS TERM HARMONICS
C *** IN THE QUASI STEADY APPROXIMATION.THESE ARE SIMPLY THE
C *** THE PRODUCTS OF THE RELEVANT PARTIAL DERIVATIVES AND THE
C *** RELEVANT FOURIER COMPONENTS OF u,v. IF NPHLAG.EQ.1 WE ALSO
C *** INCLUDE PHASE LAG IN FORCE RESPONSE BY USE OF SUBROUTINE
C *** 'PHLAG' WHICH EVALUATES SUCH A QUANTITY BY EITHER AMIET'S
C *** LOW FREQUENCY FORMULA OR BY A FRESNEL INTEGRAL BASED FORMULA.
C *** THESE CALCULATIONS APPEAR LENGTHY BECAUSE WE HAVE TO
C *** CAREFULLY PRESERVE THE RELATIVE PHASE OF EACH CONTRIBUTION.
C *** AS EXPLAINED IN THE REPORT,THIS IS DONE BY CALCULATING THE
C *** AMPLITUDE OF EACH CONTRIBUTION AS WELL FINDING THE ANGLE
C *** IN THE PHI DIRECTION AT WHICH THAT CONTRIBUTION ACHIEVES
C *** ITS MAXIMUM POSITIVE VALUE.
C *** FZ0 & FT0 ARE THE STEADY LOADING CONTRIBUTIONS
C
            FNROT = FLOAT(NROT)
            IF (ICALL.EQ.1) FNROTF = FNROT
            FZ0    = TF
            FT0    = FNROT*QF
C
C *** THE DO 90 LOOP EVALUATES THE UNSTEADY HARMONICS
C
            DO 90 I = 1,NMODES
              FI = FLOAT(I)
C
C *** CALCUALTE PHASE LAG IN SUBROUTINE PHLAG, IF REQUIRED
C
              PHLG  =  0.0
              IF(NPHLAG.EQ.1) CALL PHLAG (MT,FI,FR,LAM1,NB,SIG,PHLG)
              PHLGD = PHLG*(180./PI)/FI
C
C *** AXIAL FORCE HARMONICS
C
              A1      = PDTU*AMPU(I)
              IF(A1.LT.0.00) PHUT = PHU(I)+PI/FI
              IF(A1.GT.0.00) PHUT = PHU(I)
              IF(A1.LT.0.00) A1    = -A1
              CN1     = COS(FI*PHUT)
              SN1     = SIN(FI*PHUT)
              A2      = FNROTF*PDTV*AMPV(I)
              IF(A2.LT.0.00) PHVT = PHV(I)+PI/FI
              IF(A2.GT.0.00) PHVT = PHV(I)
              IF(A2.LT.0.00) A2    = -A2
              CN2     = COS(FI*PHVT)
              SN2     = SIN(FI*PHVT)
              CN12    = COS(FI*(PHUT-PHVT))
              PHUTD   = PHUT/DEGR
              PHVTD   = PHVT/DEGR
              AXA     = SQRT(A1**2+A2**2+2.*A1*A2*CN12)
              DEN     = A1*CN1+A2*CN2
              ADEN    = ABS(DEN)
```

```
              IF(ADEN.EQ.0.00) PHIP = .5*PI/FI
              IF(ADEN.GT.0.00) PHIP = ATAN((A1*SN1+A2*SN2)/DEN)/FI
              CNA1    = COS(FI*(PHIP-PHUT))
              CNA2    = COS(FI*(PHIP-PHVT))
              TEST    = A1*CNA1+A2*CNA2
              IF(TEST.LT.0.00) PHIP = PHIP+PI/FI
              PHIPD   = PHIP/DEGR
              ZARG    = -IM*PHIP*FI-IM*PHLG*FNROT
              Z       = CEXP(ZARG)
              ZC      = CONJG(Z)
              FZP(I) = AXA*Z
              FZM(I) = AXA*ZC
C
C *** TANGENTIAL FORCE HARMONICS
C
              A1      = FNROT*PDQU*AMPU(I)
              IF(A1.LT.0.00) PHUT = PHU(I)+PI/FI
              IF(A1.GT.0.00) PHUT = PHU(I)
              IF(A1.LT.0.00) A1   = -A1
              CN1     = COS(FI*PHUT)
              SN1     = SIN(FI*PHUT)
              A2      = FNROT*FNROTF*PDQV*AMPV(I)
              IF(A2.LT.0.00) PHVT = PHV(I)+PI/FI
              IF(A2.GT.0.00) PHVT = PHV(I)
              IF(A2.LT.0.00) A2   = -A2
              CN2     = COS(FI*PHVT)
              SN2     = SIN(FI*PHVT)
              CN12    = COS(FI*(PHUT-PHVT))
              PHUTD   = PHUT/DEGR
              PHVTD   = PHVT/DEGR
              AXA     = SQRT(A1**2+A2**2+2.*A1*A2*CN12)
              DEN     = A1*CN1+A2*CN2
              ADEN    = ABS(DEN)
              IF(ADEN.EQ.0.00) PHIP = .5*PI/FI
              IF(ADEN.GT.0.00) PHIP = ATAN((A1*SN1+A2*SN2)/DEN)/FI
              CNA1    = COS(FI*(PHIP-PHUT))
              CNA2    = COS(FI*(PHIP-PHVT))
              TEST    = A1*CNA1+A2*CNA2
              IF(TEST.LT.0.00) PHIP = PHIP+PI/FI
              PHIPD   = PHIP/DEGR
              ZARG    = -IM*PHIP*FI-IM*PHLG*FNROT
              Z       = CEXP(ZARG)
              ZC      = CONJG(Z)
              FTP(I) = AXA*Z
              FTM(I) = AXA*ZC
C
C *** UNSTEADY THICKNESS HARMONICS ***
C
              A1       = CTST*AMPU(I)
              IF(A1.LT.0.00) PHUT = PHU(I)+PI/FI
              IF(A1.GT.0.00) PHUT = PHU(I)
              IF(A1.LT.0.00) A1   = -A1
              CN1      = COS(FI*PHUT)
              SN1      = SIN(FI*PHUT)
              A2       = CT2*AMPV(I)
              IF(A2.LT.0.00) PHVT = PHV(I)+PI/FI
              IF(A2.GT.0.00) PHVT = PHV(I)
              IF(A2.LT.0.00) A2   = -A2
```

```
              CN2      = COS(FI*PHVT)
              SN2      = SIN(FI*PHVT)
              CN12     = COS(FI*(PHUT-PHVT))
              PHUTD    = PHUT/DEGR
              PHVTD    = PHVT/DEGR
              AXA      = SQRT(A1**2+A2**2+2.*A1*A2*CN12)
              DEN      = A1*CN1+A2*CN2
              ADEN     = ABS(DEN)
              IF(ADEN.EQ.0.00) PHIP = .5*PI/FI
              IF(ADEN.GT.0.00) PHIP = ATAN((A1*SN1+A2*SN2)/DEN)/FI
              CNA1     = COS(FI*(PHIP-PHUT))
              CNA2     = COS(FI*(PHIP-PHVT))
              TEST     = A1*CNA1+A2*CNA2
              IF(TEST.LT.0.00) PHIP = PHIP+PI/FI
              PHIPD    = PHIP/DEGR
              ZARG     = -IM*PHIP*FI
              Z        = CEXP(ZARG)
              ZC       = CONJG(Z)
              UTHP(I) = AXA*Z
              UTHM(I) = AXA*ZC
  90         CONTINUE
C
C *** JUMPED TO HERE IF ICOUNT IS GREATER THAN 0 ***
C
  100        CONTINUE
C
C *** NOW CALCULATE IN DO 120 LOOP,FOR EACH PHI,NOISE DUE AXIAL
C *** FORCES('CSUMZ'),NOISE DUE TO TANGENTIAL FORCES('CSUMT') AND
C *** NOISE DUE TO THICKNESS('CSUTH').'PCV(IK,J,IJKL,IRAD)' CONTAINS
C *** THE TOTAL NOISE DUE TO THE THREE SOURCES AT THE 'IK' HARMONIC,
C *** PHI ANGLE 'J',THETA ANGLE 'IJKL' AND FROM RADIAL ELEMENT
C *** CENTERED AT 'IRAD'.'PCVSL(IK,IJKL,IRAD)' LIKEWISE CONTAINS
C *** THE STEADY LOADING&THICKNESS NOISE.CLEARLY 'PCVSL' DOES NOT
C *** DEPEND ON 'PHI' OR 'J'.
C
              FZ       = FLOAT(NZERO)
              M0       = FLOAT(NZT)
              MYCON    = .5*KA*CAMP*MY
             DO 120 J = 1,NPHID
              PHD      = PHDV(J)
              PHI      = PHD*DEGR
              PHIA     = PHI-.5*PI
              CSUMZ    = BF0*FZ0
              CSUMT    = BF0*FT0*FLOAT(NZT)
              IF(J.EQ.1)CZSL = CSUMZ
              IF(J.EQ.1)CTSL = CSUMT
              CPHI     = COS(PHI)
              SPHI     = SIN(PHI)
              MYPHI    = MY*SPHI
              CTRM     = MYPHI*(MFLT+CTHETM)*CAMP
              CPSI     = CTHETM+CTRM*STHETM
              SPSI     = STHETM-CTRM*CTHETM
              RBMOD    = 1./(1.-MYPHI*STHETM*CAMP)
              CMPMOD = (1./(1.+MFLT*CPSI+MYPHI*STHETM))/CAMP
              PHMOD    = (1.,0.)
              KACMOD   = CPSI*CMPMOD/CTHETM
              KASY     = KAS*CMPMOD*(SPSI+MYPHI)/STHETM
              CALL BJFNA (KASY,NBMAX,BM)
```

```
                BF0Y     = BM(NZERO+1)
                IF((NZT.LT.0).AND.(NZTD.NE.0))BF0Y = -BF0Y
                CSUTH    = BF0Y*PHMOD*RBMOD*CMPMOD**3
                CSUMZ    = BF0Y*FZ0*KACMOD*PHMOD*RBMOD*CMPMOD
                CSUMT    = BF0Y*FT0*PHMOD*RBMOD*CMPMOD*FLOAT(NZT)
                DO 110 I = 1,NMODES
                  FI     = FLOAT(I)
                  MP     = FLOAT(NZT+I)
                  MM     = FLOAT(NZT-I)
                  Z      = CEXP(IM*FI*PHIA)
                  ZC     = CONJG(Z)
                  ZP     = Z*BFP(I)
                  ZM     = ZC*BFM(I)
C
C-------------------------------------------------------------------
C
C THE GE. EVENDALE H6000 HONEYWELL COMPUTER SYSTEM WAS USED TO DEBUG
C CHECK OUT THIS PROGRAM.  THE HONEYWELL'S NUMERIC RANGE IS APPROX.
C 10E+38 AND 10E-37.
C IT WAS DESIRED TO INCLUDE THE FOLLOWING 10 LINES OF CODE BECAUSE
C THE VALUES OF ZP AND ZM WERE EXCEEDING THE LOWER LIMIT OF THE
C HONEWELL'S NUMERIC RANGE AND ISSUING EXP. UNDERFLOW ERROR MESSEAGES.
C THE ACCEPTED SOLUTION WAS TO SET EITHER PART (REAL OR IMAGINARY) OF
C THE COMPLEX VALUES ZP OR ZM TO 0.0 IF THAT PART WAS LESS THAN 1E-20.
C THESE 10 LINES OF CODE MAY BE COMMENTED OUT OR DELETED IF THIS
C PROGRAM IS RUN ON A COMPUTER WITH GREATER NUMERIC RANGE.
C
        CNRP=REAL(ZP)
        CNIP=AIMAG(ZP)
        IF(ABS(CNRP).LT.1E-20)CNRP=0.0
        IF(ABS(CNIP).LT.1E-20)CNIP=0.0
        ZP=CMPLX(CNRP,CNIP)
        CNRP=REAL(ZM)
        CNIP=AIMAG(ZM)
        IF(ABS(CNRP).LT.1E-20)CNRP=0.0
        IF(ABS(CNIP).LT.1E-20)CNIP=0.0
        ZM=CMPLX(CNRP,CNIP)
C
C-------------------------------------------------------------
C
                  IF(I.GT.1) GO TO 105
                  FZ0T=MYCON*FZ0
                  FT0T=MYCON*FT0*M0
                  CSUMZ = CSUMZ+ZP*(FZP(I)+FZ0T)+ZM*(FZM(I)-FZ0T)
                  CSUMT = CSUMT+ZP*(FTP(I)*MP+FT0T)+ZM*(FTM(I)*MM
     &                                              -FT0T)
                  CSUTH = CSUTH+ZP*UTHP(I)*TPLV(I)+ZM*UTHM(I)*TMINV(I)
     &                                 +ZP*MYCON-ZM*MYCON
                  GO TO 110
 105              CONTINUE
                  CSUMZ = CSUMZ+ZP*FZP(I)+ZM*FZM(I)
                  CSUMT = CSUMT+ZP*FTP(I)*MP+ZM*FTM(I)*MM
                  CSUTH = CSUTH+ZP*UTHP(I)*TPLV(I)+ZM*UTHM(I)*TMINV(I)
 110            CONTINUE
                CSUMZ    = KAC*CSUMZ
                IF(J.EQ.1)CZSL = KAC*CZSL
                PCV(IK,J,IJKL,IRAD) = AC*(CSUMT+CSUMZ)*RADPH
     &                                +CSUTH*RADPH*CCTHCK
```

```
              IF(J.EQ.1)PCVSL(IK,IJKL,IRAD) = AC*(CTSL+CZSL)*RADPH
     &                                        +BF0*RADPH*CCTHCK
 120          CONTINUE
              ICOUNT = ICOUNT+1
 130        CONTINUE
 140      CONTINUE
 150    CONTINUE
C
C ****************************************
C *   ACOUSTIC CALCULATIONS COMPLETED    *
C ****************************************
C
C *** IN THE DO 190 LOOP,WE ADD TOGETHER CONTRIBUTIONS FROM ALL RADII,
C *** COMPUTE AMPLITUDES,SWITCH TO DB AND ALSO COMPUTE PHYSICAL DB
C
      REFDB = 127.58+20*ALOG10(PO)
      DO 190 I = 1,NHMAX
        DO 180 J = 1,NPHID
          DO 170 K = 1,NTHTDI
            CSUMP = (0.00,0.00)
            IF(J.EQ.1)CSMPSL = (0.00,0.00)
            DO 160 IRAD = 1,NRAD
              CSUMP = CSUMP+PCV(I,J,K,IRAD)
              IF(J.EQ.1)CSMPSL = CSMPSL+PCVSL(I,K,IRAD)
 160        CONTINUE
            PN   = CABS(CSUMP)
            IF(J.EQ.1)PNSL = CABS(CSMPSL)
            PDB = REFDB+20.*ALOG10(PN)
            IF(PDB.LT.0.) PDB = 0.
            PDBV(I,J,K,ICALL) = PDB
            IF(J.EQ.1) THEN
              PDBSTL = REFDB+20.*ALOG10(PNSL)
              IF(PDBSTL.LT.0.) PDBSTL = 0.
              PDBSL(I,K,ICALL) = PDBSTL
            END IF
 170      CONTINUE
 180    CONTINUE
 190  CONTINUE
C
C *** THE DO 220 LOOP COMPUTES THE INSTALLATION EFFECT AS 'DIFFDB(I,K,J)
C ***   WHERE:  'I' GOES WITH HARMONIC #,
C ***           'J' GOES WITH 'PHI'
C ***           'K' GOES WITH 'THETA'.
C
      DO 220 I = 1,NHMAX
        DO 210 J = 1,NPHID
          DO 200 K = 1,NTHTDI
            DIFFDB(I,J,K,ICALL) = PDBV(I,J,K,ICALL)-PDBSL(I,K,ICALL)
 200      CONTINUE
 210    CONTINUE
 220  CONTINUE
C
C *** OUTPUT THE RESULTS
C
        IFINAL = 0
      CALL OUTPUT (NPHLAG,NHMAX,NANG,NTHTDI,NPHID,ICALL,THTDIV,PHDV,
     & DIFFDB,PDBSL,ARAD,IFINAL)
C
```

```
C *** COMBINE ISOLATED AND INSTALLED RESULTS
C
        IF(ISOLAT.EQ.1.AND.ICALL.EQ.2) THEN
            DO 260 IC=1,2
              DO 250 K=1,NTHTDI
                DO 240 J=1,NPHID
                  DO 230 I=1,NHMAX
                    DIFFDB(I,J,K,IC) = DIFFDB(I,J,K,IC)+BPHM(I,K,IC)
230               CONTINUE
240             CONTINUE
250           CONTINUE
C
              IFINAL = 1
              CALL OUTPUT (NPHLAG,NHMAX,NANG,NTHTDI,NPHID,IC,THTDIV,PHDV,
     & DIFFDB,PDBSL,ARAD,IFINAL)
C
260         CONTINUE
        END IF
 9999 CONTINUE
C
C -------------------------------------------------------------------
C
      RETURN
      END
C
C ******************************
C *      SUBROUTINE OUTPUT      *
C *********************************************************************
C
C     SUBROUTINE TO WRITE OUT THE FINAL RESULTS OF THE PREDICTED
C     INSTALLATION EFFECTS.  THE EFFECTS ARE REPRESENTED ACOUSTICALLY
C     BY DELTA dB'S AND ARE PRESENTED IN TABLE FORM FOR EACH HARMONIC
C     AS DELTA dB vs. ANGLE THETA AND ANGLE PHI. FIRST THE RESULTS
C     FOR THE FRONT ROTOR ARE WRITTEN AND NEXT THE RESULTS FOR
C     THE AFT ROTOR .
C
C *********************************************************************
C
      SUBROUTINE OUTPUT (NPHLAG,NHMAX,NANG,NTHTDI,NPHID,
     &                  ICALL,THTDIV,PHDV,DIFFDB,PDBSL,ARAD,IFINAL)
C
      DIMENSION THTDIV(20,2),PHDV(30),DIFFDB(5,30,20,2),PDBSL(5,20,2)
C
        COMMON /CRINST/ BPHM(5,17,2)
C
      DO 20 I = 1,NHMAX
        IF(IFINAL.EQ.0) THEN
            WRITE(14,1000)
        ELSE
            WRITE(14,1005)
        END IF
        IF(ICALL.EQ.1)WRITE(14,1010)
        IF(ICALL.EQ.2)WRITE(14,1020)
        IF(NPHLAG.EQ.0)WRITE(14,1030)
        IF(NPHLAG.EQ.1)WRITE(14,1040)
        IF(IFINAL.EQ.0) THEN
            WRITE(14,1050)I,ARAD
        ELSE
```

```
            WRITE(14,1055)I,ARAD
        END IF
        IF(NANG.EQ.1)WRITE(14,1060)
        IF(NANG.EQ.0)WRITE(14,1070)
        WRITE(14,1080)(THTDIV(K,ICALL),K = 1,NTHTDI)
        IF(IFINAL.EQ.0) THEN
            WRITE(14,1090)(PDBSL(I,K,ICALL),K=1,NTHTDI)
            WRITE(14,1100)
        ELSE
            WRITE(14,1090)(BPHM(I,K,ICALL),K=1,NTHTDI)
            WRITE(14,1105)
        END IF
        DO 10 J = 1,NPHID
          WRITE(14,1110)PHDV(J),(DIFFDB(I,J,K,ICALL),K = 1,NTHTDI)
 10     CONTINUE
 20   CONTINUE
C
C --- FORMAT STATEMENTS ----------------------------------------------
C
 1000 FORMAT('1',//10X,'* *   A C O U S T I C   R E S U L T S   * *')
 1005 FORMAT('1',//10X,'* *   F I N A L   C O N S O L I D A T E D  ',
     &        'A C O U S T I C   R E S U L T S   * *')
 1010 FORMAT(//20X,'   FRONT   ROTOR')
 1020 FORMAT(//20X,'   REAR    ROTOR')
 1030 FORMAT(//2X,'PHASE LAG NOT INCLUDED IN RESULTS')
 1040 FORMAT(//2X,'PHASE LAG INCLUDED IN RESULTS')
 1050 FORMAT(//2X,'HARMONIC # = ',I3,20X,F6.1,' FT. ARC PREDICTIONS',//)
 1055 FORMAT(//2X,'HARMONIC # = ',I3,20X,F6.1,' FT. SIDELINE',//)
 1060 FORMAT(/30X,'OBSERVER ANGLES from INLET re: PROPELLER AXIS')
 1070 FORMAT(/30X,'EMISSION ANGLES from INLET re: PROPELLER AXIS')
 1080 FORMAT(9X,17F6.1)
 1090 FORMAT(/1X,'REF dB>',1X,17F6.1)
 1100 FORMAT(/3X,'PHID',23X,'CRP INSTALLATION EFFECT DELTA dB ',
     & 're: Ref dB')
 1105 FORMAT(/3X,'PHID',23X,'CRP INSTALLED TONE SPL, dB ')
 1110 FORMAT(1X,F6.1,2X,17F6.1)
C
C -------------------------------------------------------------------
C
      RETURN
      END
C ********************************
C *      SUBROUTINE FCOEFF       *
C *******************************************************************
C
C     SUBROUTINE FCOEFF EVALUATES u,v DISTORTIONS AT PLANE OF
C     PROPELLER DISK DUE TO ANGLE OF ATTACK,WING+TRAILING VORTEX
C     SYSTEM,PYLON AND FUSELAGE AND ALSO CARRIES OUT A FOURIER ANALYSIS
C     OF THESE DISTORTIONS IN THE PHI DIRECTION. IT CALLS SUBROUTINE
C     FUSEL AND IMSL SUBROUTINE FFTRF.
C
C *******************************************************************
C
      SUBROUTINE FCOEFF(RAD)
C
      COMMON /H/ FRAT,FLEN1,SRAT,FRAW,EMFUS,AFUS,IRAD
C
      COMMON /G/ LAM1,CLAC,ARA,AAT,DDP,NPG,XDG,YDG,ZDG,PHU(51),
```

```
     &             PHV(51),AMPU(51),AMPV(51)
C
      COMMON /PYLON/ APYLVW(51,51),PHPLVW(51,51),APYLCU(51,51),
     &               PHPLCU(51,51),APYLCV(51,51),PHPLCV(51,51),
     &               ARBDU
C
      DIMENSION AV(100),TV(100),SA(50),CA(50),ST(50),CT(50),PHID(100)
C
      DIMENSION COEFA(100),COEFT(100)
C
      REAL KBC,LAM1
C
      COMPLEX Z1,Z2C,Z3,ZSUM,IM
C
      PI   = 3.14159265
      DEGR = PI/180.
      IM   = (0.,1.)
C
      IF(IRAD.GT.1)GO TO 10
        ADR = LAM1
C
C *** PARAMETERS RELATED TO WING/TRAILING VORTEX INDUCED FLOW
C
        CL = CLAC
        AR = ARA
        AA = AAT
        DD = DDP
        NP = NPG
        XD = XDG
        YD = YDG
        ZD = ZDG
C
C *** CF IS "s/s'" OF REPORT
C
        CF  = 4./PI
        CF2 = CF**2
C
C *** NEXT TWO LINES:UPWASH DUE TO ANGLE OF ATTACK
C
        AL    = AA*DEGR
        SAL   = SIN(AL)
        X     = CF*XD
        Y0    = CF*YD
        Z0    = ZD*CF
        FLEN1 = FRAT*CF
        KBC   = CF2*CL/(AR*4.*PI)
C
C *** ALONG WITH FLEN1,CALLING 'FUSEL' YIELDS PARAMETERS OF
C *** FUSELAGE INDUCED FLOW
C
        CALL FUSEL
C
        XF    = X-.5*(1.-2.*FRAW)*FLEN1
        XFPA  = XF+AFUS
        XFPA2 = XFPA**2
        XFMA  = XF-AFUS
        XFMA2 = XFMA**2
        DPHI  = 2.*PI/FLOAT(NP)
```

```
C
   10 CONTINUE
C
C *** RADIUS OF CIRCLE OF ANALYSIS IN PROPELLER DISK
C
      A = .5*CF*DD*RAD
      DO 20 I = 1,NP
        PHI     = DPHI*FLOAT(I-1)
        PHID(I) = PHI/DEGR
        CPHI    = COS(PHI)
        SPHI    = SIN(PHI)
        Y       = Y0-A*CPHI
        Z       = Z0+A*SPHI
C
C *** (XF,Y,Z) ARE COORDS. OF TYPICAL POINT SUITABLE FOR FSLG.
C *** (X,Y,Z) LIKEWISE FOR HORSESHOE VORTEX FLOW
C
        Z2   = Z**2
        R2   = Y**2+Z2
        R    = SQRT(R2)
        T32P = (R2+XFPA2)**1.5
        T32M = (R2+XFMA2)**1.5
C
C *** (QX,QY,QZ)-FSLG. INDUCED FLOW
C
        QX   = EMFUS*(XFPA/T32P-XFMA/T32M)
        QR   = EMFUS*(1./T32P-1./T32M)*R
        QY   = QR*Y/R
        QZ   = QR*Z/R
C
C *** SEE PAGES 158-159 OF GLAUERT:'AIRFOIL & AIRSCREW THEORY'
C *** WE ARE CALCULATING (u1,v1,w1)&(u2,v2,w2)&(u3,v3,w3) BY
C *** HIS FORMULAE
C
        YPS  = Y+1.
        YMS  = Y-1.
        YPS2 = YPS**2
        YMS2 = YMS**2
        TXZ  = X**2+Z2
        TZYP = Z2+YPS2
        TZYM = Z2+YMS2
        TP   = SQRT(TXZ+YPS2)
        TM   = SQRT(TXZ+YMS2)
        T1   = Z/TXZ
        T2   = YPS/TP
        T3   = YMS/TM
        T23  = T2-T3
        T4   = X/TXZ
        T5   = Z/TZYM
        T6   = X/TM
        T61  = 1.+T6
        T7   = YMS/TZYM
        T8   = Z/TZYP
        T9   = X/TP
        T91  = 1.+T9
        T10  = YPS/TZYP
        U1   =  T1*T23
        W1   = -T4*T23
```

```
      V2    = -T5*T61
      W2    =  T7*T61
      V3    =  T8*T91
      W3    = -T10*T91
C
C *** NOW CALCULATE (u,v,w) BY ADDING PREVIOUS RESULTS
C
      U     = U1
      V     = V2+V3
      W     = W1+W2+W3
C
C *** 'AV' IS TOTAL AXIAL VELOCITY DISTORTION
C *** 'TV' IS TOTAL TANGENTIAL VELOCITY DISTORTION
C
      AV(I) = KBC*U+QX-ARBDU*SPHI
      TV(I) = KBC*(W*CPHI+V*SPHI)+(SAL+QZ)*CPHI+QY*SPHI
C
 20   CONTINUE
C
C *** FOURIER DECOMPOSITION OF AV,TV
C
C *** CALL IMSL SUBROUTINE TWICE
C
      CALL FFTRF (NP,AV,COEFA)
      CALL FFTRF (NP,TV,COEFT)
      NK    = NP/2
      CA(1) = 2.*COEFA(1)
      CT(1) = 2.*COEFT(1)
      SA(1) = 0.00
      ST(1) = 0.00
      SA(NK+1) = 0.00
      ST(NK+1) = 0.00
      DO 25 I = 2,NP
        ICRIT = MOD(I,2)
        IF (ICRIT.EQ.0)      IV = I/2+1
        IF (ICRIT.NE.0)      IV = (I+1)/2
        IF (ICRIT.EQ.0) CA(IV) =  2.*COEFA(I)
        IF ((ICRIT.EQ.0).AND.(I.EQ.NP)) CA(IV) = .5*CA(IV)
        IF (ICRIT.NE.0) SA(IV) = -2.*COEFA(I)
        IF (ICRIT.EQ.0) CT(IV) =  2.*COEFT(I)
        IF ((ICRIT.EQ.0).AND.(I.EQ.NP)) CT(IV) = .5*CT(IV)
        IF (ICRIT.NE.0) ST(IV) = -2.*COEFT(I)
 25   CONTINUE
C
C *** SORT OUT COEFFICIENTS INTO AMPL. & ANGLE OF OCCURRENCE
C *** OF MAXIMUM POSITIVE VALUE.ALSO SOME NORMALISATION.
C *** ADD IN PYLON CONTRIBUTIONS
C
      DO 30 K = 1,NK
        KK      = K+1
        AMPA    = SQRT(SA(KK)**2+CA(KK)**2)/FLOAT(NP)
        CK      = ABS(CA(KK))
        IF(CK.EQ.0.00) PHA = .5*PI/FLOAT(K)
        IF(CK.GT.0.00) PHA = ATAN(SA(KK)/CA(KK))/FLOAT(K)
        SPHA    = SIN(PHA)
        CPHA    = COS(PHA)
        TEST    = CA(KK)*CPHA+SA(KK)*SPHA
        IF(TEST.LT.0.0) PHA = PHA+PI/FLOAT(K)
```

```
         PHAD    = PHA/DEGR
         AMPU(K) = AMPA*.5
         PHU(K)  = PHA
         AMPT    = SQRT(ST(KK)**2+CT(KK)**2)/FLOAT(NP)
         CK      = ABS(CT(KK))
         IF(CK.EQ.0.00) PHT = .5*PI/FLOAT(K)
         IF(CK.GT.0.00) PHT = ATAN(ST(KK)/CT(KK))/FLOAT(K)
         SPHT    = SIN(PHT)
         CPHT    = COS(PHT)
         TEST    = CT(KK)*CPHT+ST(KK)*SPHT
         IF(TEST.LT.0.0) PHT = PHT+PI/FLOAT(K)
         PHTD    = PHT/DEGR
         AMPV(K) = AMPT*.5
         PHV(K)  = PHT
         FK      = FLOAT(K)
         Z1      = AMPU(K)*CEXP(-IM*FK*PHU(K))
         Z2C     = APYLVW(K,IRAD)*CEXP(-IM*FK*PHPLVW(K,IRAD))
         Z3      = APYLCU(K,IRAD)*CEXP(-IM*FK*PHPLCU(K,IRAD))
         ZSUM    = Z1+Z2C+Z3
         AMPU(K) = CABS(ZSUM)
         IF(AMPU(K).GT.0.00) PHU(K) = ATAN2(-AIMAG(ZSUM),REAL(ZSUM))
      &                                /FK
         Z1      = AMPV(K)*CEXP(-IM*FK*PHV(K))
         Z3      = APYLCV(K,IRAD)*CEXP(-IM*FK*PHPLCV(K,IRAD))
         ZSUM    = Z1+Z3
         AMPV(K) = CABS(ZSUM)
         IF(AMPV(K).GT.0.00) PHV(K) = ATAN2(-AIMAG(ZSUM),REAL(ZSUM))
      &                                /FK
 30   CONTINUE
C
      RETURN
      END
C
C *****************************
C *      SUBROUTINE FUSEL      *
C ***************************************************************************
C
C     COMPUTES PARAMETERS NEEDED TO EVALUATE FUSELAGE INDUCED FLOW
C     BASED ON RANKINE SOLID MODEL. IT CALLS IMSL SUBROUTINE NEQNF.
C
C ***************************************************************************
C
      SUBROUTINE FUSEL
C
      COMMON /H/ FRAT,FLEN1,SRAT,FRAW,EMFUS,AFUS,IRAD
C
      COMMON /S/ L
C
      REAL L,X(2),XGUESS(2)
C
      EXTERNAL FCNFU
C
      N      = 2
      ERRREL = .0001
      ITMAX  = 200
      L      = FLEN1
      XGUESS(1) = .5*L
      XGUESS(2) = .5*(.5*SRAT*L)**2
```

```
C
C *** CALL IMSL SUBROUTINE
C
      CALL NEQNF (FCNFU,ERRREL,N,ITMAX,XGUESS,X,FNORM)
C
      SSTR  = .5*X(2)
      SLEN  = 2.*X(1)
      EMFUS = SSTR
      AFUS  = X(1)
C
      RETURN
      END
C
C *************************************
C *      FUNCTION FCNFU SUBROUTINE      *
C ***********************************************************************
C
C     FUNCTION WHOSE ZERO IS TO BE FOUND TO EVALUATE FUSELAGE
C     INDUCED FLOW BASED ON RANKINE SOLID MODEL.
C
C ***********************************************************************
C
      SUBROUTINE FCNFU(X,F,N)
C
      DIMENSION X(2),F(2)
C
      COMMON /H/ FRAT,FLEN1,SRAT,FRAW,EMFUS,AFUS,IRAD
C
      COMMON /S/ L
C
      REAL L
C
      EL   = .5*L
      A    = X(1)
      A2   = A**2
      B2   = X(2)
      H    = .5*SRAT*L
      EL2  = EL**2
      H2   = H**2
      F(1) = (EL2-A2)**2-2.*A*B2*EL
      F(2) = 2.*A*B2-H2*SQRT(H2+A2)
C
      RETURN
      END
C
C ******************************
C *      SUBROUTINE PHLAG      *
C ***********************************************************************
C
C     TWO DIMENSIONAL,LINEARISED,COMPRESSIBLE,FLAT PLATE
C     CONVECTED GUST PHASE LAG FORMULAE BASED ON AMIET'S
C     LOW FREQUENCY THEORY WHERE APPLICABLE AND ON A
C     FRESNEL INTEGRAL BASED FORMULA OTHERWISE.
C     SEE GOLDSTEIN:'AEROACOUSTICS':PAGES 138-139,EQS.(3.70)
C     AND (3.71) FOR THESE FORMULAE . IT CALLS SUBROUTINE BJFNA
C     AND USES IMSL FUNCTION PROGRAM CERFE .
C
C ***********************************************************************
```

```
C
      SUBROUTINE PHLAG(MT,FI,FR,LAM1,NB,SIG,PHLG)
C
      COMPLEX ZARG,W,IM,SS,SC,Z,ZARG2,FRNI
C
      DOUBLE PRECISION BARG,BF(2)
C
      REAL MT,MR,LAM1,MA,MTR
C
      PI   = 3.1415926535
      MTR  = MT*FR
      MA   = LAM1*MT
      MR   = SQRT(MA**2+MTR**2)
      IM   = (0.,1.)
      SIG1 = FI*(MTR/MR)*PI*SIG/FLOAT(NB)
      BR2  = 1.-MR**2
      BR   = SQRT(ABS(BR2))
C
C *** 'PAR' AND 'MR' DECIDE CHOICE OF AMIET OR FRESNEL
C
      PAR = ABS(SIG1*MR/BR2)
      IF((PAR.GT.1.).OR.(MR.GE.1))GO TO 10
C
C *** AMIET FORMULA
C
        SIG1S = SIG1/BR2
        T3    = (1.+2.*PI*SIG1S)
        T1    = -SIG1S*(1.-PI**2*.5/T3)
        T2    = SQRT(T3)
        SS    = CEXP(T1*IM)/T2
        BARG  = MR*PAR
C
C *** CALCULATE BESSEL FUNCTIONS
C
        CALL BJFNA (BARG,2,BF)
C
        FMR = (1.-BR)*ALOG(MR)+BR*ALOG(1.+BR)-ALOG(2.)
        T1  = -SIG1*FMR/BR2
        SC  = SS*(BF(1)+IM*BF(2))*CEXP(IM*T1)/BR
        GO TO 20
C
C *** FRESNEL INTEGRAL FORMULA
C
 10   CONTINUE
C
      Z          = CEXP(-IM*SIG1)*(1.+IM)/(SIG1*PI*SQRT(MR))
      ARG        = SQRT((4.*SIG1*MR)/(PI*(1.+MR)))
      ZARG       = SQRT(PI)*.5*ARG*(1.+IM)
      ZARG2      = ZARG**2
C
C *** USE IMSL FUNCTION PROGRAM
C
      W    = CERFE (ZARG)
C
      FRNI = (1.+IM)*.5*(1.-CEXP(ZARG2)*W)
      SC   = Z*FRNI
C
 20   CONTINUE
```

```
C
      PHLG = ATAN2(AIMAG(SC),REAL(SC))
C
      RETURN
      END
C
C ******************************
C *      SUBROUTINE LSPFIT      *
C ***********************************************************************
C
C     THIS ROUTINE IS A LEAST-SQUARES PARABOLIC CURVE FIT PROGRAM THAT
C     WILL INTEGRATE OR INTERPOLATE USING A PARABOLA WHICH PASSES THROUGH
C     POINTS I AND I+1 AND MISSES POINTS I-1 AND I+2 (IF THEY BOTH
C     EXIST) SUCH THAT THE SQUARE OF THE DEVIATION IS A MINIMUM.
C     NOTE THAT  I  IS GENERALLY SELECTED SUCH THAT
C                 X(I).LE.XC.LT.X(I+1)
C     THE EQUATION FOR THE PARABOLA IS:
C                 Y-Y(I)  =  B*(X-X(I)) + C*(X-X(I))**2
C     OUTSIDE OF THE X,Y-DATA RANGE, LINEAR EXTRAPOLATION OF THE
C     PARABOLA END POINT SLOPE IS EMPLOYED IF NXTRP = 0......29 AUG 85
C
C     INPUT:
C       X, Y      = PTS. ON CURVE
C       NPTS      = NO. OF X    (NPTS > 0)
C       XC        = LIST OF X AT WHICH CALC TO BE DONE
C       YC(1)     = INTEGRATION CONSTANT IF  ND = -1
C       NXC       =  NO. OF XC
C       ND        = 0 TO GET COORD,    = 1 TO GET 1ST DERIVATIVE,
C                 = -1 FOR INTEGRATION    (NPTS > 2)
C       LEND      =  LINEAR FIT IN END INTERVAL, T OR F
C       NXTRP = T OR F TO USE Y-TABLE END VALUES OR EXTRAPOLATE
C
C     OUTPUT:
C       YC        = COORDINATE OR DERIVATIVE AT XC     OR
C       YC(IC)    = INTEGRAL(Y*DX) FROM XC(1) TO XC(IC) WHERE IC = 2,NXC
C
C     NOTES
C          'X' MAY BE IN EITHER ASCENDING OR DESCENDING ORDER.
C          FOR INTEGRATION 'XC' MUST BE IN THE SAME ORDER AS 'X';
C          FOR INTERPOLATION, NO SPECIAL ORDER IS REQIRED.
C
C **********************************************************************
C
      SUBROUTINE LSPFIT(X,Y,NPTS, XC,YC,NXC, ND)
C
      LOGICAL             LEND
      LOGICAL          NXTRP
      LOGICAL          WITHIN
C
      DIMENSION X(1),Y(1), XC(1),YC(1)
C
      COMMON /CLSPF / INVL,LEND
      COMMON /CLSPF2/ NXTRP
C
      DATA INVL,LEND,NXTRP/0,.FALSE.,.FALSE./
C
      N     = NPTS-1
      IF(ND.EQ.(-1)) INVL = 0
```

```
      ISAVE = -1
      SGN   = SIGN(1.,X(N+1)-X(1))
C
C *** BEGIN LOOP FOR EVALUATING YC(IC) AT XC(IC),  IC = 1,NXC
C
      IC    = 1
C
C *** LOCATE APPROPRIATE INTERVAL, INVL
C ***
C *** I     = INTERVAL INDEX WITHIN DATA LIMITS  (1....N)
C *** INVL  = EXPANDED INTERVAL INDEX  (0....N+1),
C ***   WHERE BY IMPLICATION THE ZEROTH INTERVAL IS
C ***   FROM  -INFINITY TO X(1)  AND  THE N+1 INTERVAL IS
C ***   FROM  X(NPTS) TO +INFINITY.  HENCE, AN INTERVAL, INVL,
C ***   CAN ALWAYS BE FOUND WHICH BOUNDS  XC(IC).
C
 100  INVL   = MAX(0,MIN(INVL,N+1))
      WITHIN = .FALSE.
      NCOUNT = N+2
 110  IF(NCOUNT) 200,120,120
 120  NCOUNT = NCOUNT-1
C
C *** CHECK FOR JUST ONE INPUT POINT
C *** IF(N.EQ.0)      GO TO 220
C *** CHECK IF XC IS TO THE RIGHT OF THE INTERVAL'S LEFT BOUND
C
      IF(INVL.EQ.0)  GO TO 160
      XI = X(INVL)
      XD = XC(IC)-XI
 130  IF(SGN*XD) 140,150,160
C
C *** F.LT.0. (F IS THE FRACTIONAL POSITION IN THE INTERVAL)
C
 140  IF(ND.EQ.(-1)) GO TO 200
      INVL = INVL-1
      GO TO 110
C
C *** F.EQ.0
C *** IF(X(INVL+1).NE.XI) GO TO 210 GGA 10/09/85 DONT CHECK IF BEYOND END
C
 150  IF(INVL.GE.NPTS.OR.X(INVL+1).NE.XI) GO TO 210
      IF(INVL.GT.N) GO TO 140
      GO TO 190
C
C *** F.GT.0.
C *** CHECK IF XC IS TO THE LEFT OF THE INTERVAL'S RIGHT BOUND
C
 160  IF(INVL.EQ.NPTS) GO TO 210
      IF(SGN*(XC(IC)-X(INVL+1))) 210,170,180
C
C *** F.EQ.1.0,  CHK FOR INTEGRATION AND DOUBLE PT BEFOR INCREMENTING INVL
C
 170  IF( (ND.EQ.(-1)) .OR.
     & (INVL.NE.N .AND. X(INVL+1).EQ.X(INVL+2)) ) GO TO 210
C
C *** F.GT.1.0
C
 180  IF(ND.EQ.(-1)) GO TO 220
```

```
 190  INVL = INVL+1
      GO TO 110
C
C *** UNSUCCESSFUL SEARCH FOR APPROPRIATE INTERVAL
C
 200  CONTINUE
C
 210  WITHIN = .TRUE.
C
C *** LEAST SQUARE PARABOLA FIT  FOR INTERVAL INVL
C
 220  IF(INVL-ISAVE) 230,300,230
 230  ISAVE = INVL
      I     = MAX(1,MIN(INVL,N))
      XI    = X(I)
      XD    = XC(IC)-XI
      YI    = Y(I)
      B     = 0.
      C     = 0.
      TOP   = 0.
      BOT   = 0.
      IF(N.EQ.0) GO TO 300
      X3    = X(I+1)-XI
      Y3    = Y(I+1)-YI
      IF(LEND .AND. (I.EQ.1 .OR. I.EQ.N)) GO TO 270
      IF(I.EQ.1) GO TO 240
      X1    = X(I-1)-XI
      X13   = X(I-1)-X(I+1)
      TOP   = X1*(Y3*X1-(Y(I-1)-YI)*X3)*X13
      BOT   = (X1*X13)*(X1*X13)*X3
 240   IF(I.GE.N) GO TO 260
      IF(XD.EQ.0. .AND. BOT.NE.0.) GO TO 250
      X4    = X(I+2)-XI
      X43   = X(I+2)-X(I+1)
      Y4    = Y(I+2)-YI
      TOP   = TOP + X4*(Y3*X4-Y4*X3)*X43
      BOT   = BOT + (X4*X43)*(X4*X43)*X3
      GO TO 260
 250  ISAVE = 0
C
C *** (X1**2 + X43**2)  MUST BE GREATER THAN  (X3/1000)**2
C
 260  IF(BOT.NE.0. .AND. ABS(BOT).GE.ABS((X3*X3)*(X3*X3)*X3*1.E-6))
     & C    = -TOP/BOT
 270   IF(X3.NE.0.) B = (Y(I+1)-YI)/X3 - C*X3
C
C *** IF XC IS OUTSIDE OF X-RANGE, REDEFINE CURVE PARAMETERS
C *** TO EXTRAPOLATE AT END POINT SLOPE.
C
      IF(I.EQ.INVL) GO TO 300
      IF(INVL.EQ.0) GO TO 290
C
C *** RIGHT END EXTRAPOLATION, ADJUST B, YI AND XD
C
      B  = B+2*C*X3
      YI = Y(I+1)
      XI = X(I+1)
 290  C      =  0.
```

```
      IF(NXTRP) B = 0.
C
C *** INTEGRATION, INTERPOLATION OR DIFFERENTIATION, STORE RESULT IN YC(IC)
C
 300  XD = XC(IC)-XI
      IF(ND) 310,340,350
C
C *** ND = -1, INTEGRATE
C
 310  IF(.NOT.WITHIN) XD = X(INVL+1)-XI
      S1 = (YI + (B/2. + C/3.*XD)*XD)*XD
      IF(WITHIN) GO TO 320
C
C *** 'INVL' IS BEING INCREMENTED TO FIND APPROPRIATE INTERVAL.  HENCE,
C *** CUMULATE THE INTEGRAL OF THE THIS INTERVAL.
C
      SA = SA + S1
      GO TO 190
C
C *** APPROPRIATE INTERVAL FOUND.   X(INVL)-XC(IC)-X(INVL+1)
C
 320  IF(IC.NE.1) GO TO 330
C
C *** LEFT LIMIT OF INTEGRATION
C
      SA = YC(IC)-S1
      GO TO 360
C
C *** RIGHT LIMIT OF INTEGRATION
C
 330  YC(IC) = SA+S1
      GO TO 360
C
C *** ND = 0,  INTERPOLATE FOR COORDINATES
C
 340  YC(IC) =  YI + (B + C*XD)*XD
      GO TO 360
C
C *** ND = 1,  FIRST DERIVATIVE
C
 350  YC(IC) =  B + 2.*C*XD
      GO TO 360
C
C *** END OF LOOP, INDEX IC
C
 360  IC = IC+1
      IF(NXC-IC) 380,370,370
 370  IF(ND.NE.(-1).AND.XC(IC).EQ.XC(IC-1)) INVL = INVL+1
      GO TO 100
C
 380  CONTINUE
C
      RETURN
      END
C
C ********************************
C *      SUBROUTINE BJFNA       *
C ***********************************************************************
```

```
C
C     CALCULATE BESSEL FUNCTIONS OF THE FIRST KIND OF NON-NEGATIVE
C     INTEGER ORDER FOR REAL ARGUMENTS USING RECURSION METHOD.
C
C     INPUT:  X  = ARGUMENT VALUE
C             N  = NUMBER OF FUNCTION VALUES TO BE COMPUTED
C
C     OUTPUT: BJ = VECTOR OF LENGTH N CONTAINING THE BESSEL FUNCTIONS
C                  FOR ORDERS 0 THRU N-1.
C
C                  EG.: BJ(1) CONTAINS THE ORDER 0 FUNCTION
C                       BJ(2) CONTAINS THE ORDER 1 FUNCTION
C                         .
C                         .
C                       BJ(N) CONTAINS THE ORDER N-1 FUNCTION
C
C **********************************************************************
C
      SUBROUTINE BJFNA(X,N,BJ)
C
      DOUBLE PRECISION BJ(N),BESSEL(502)
C
C *** ZERO OUT BJ ARRAY VECTOR
C
      DO 10 I=1,N
        BJ(N)=0.0
 10   CONTINUE
C
C *** DETERMINE PRACTICAL LIMIT OF ORDER N WHERE FUNCTION VALUE IS ZERO
C
      IF(N.GE.100)GO TO 20
        NLIM=-.0025494*X**2+1.448052*X+14.
        GO TO 30
 20   CONTINUE
      NLIM=1.06*X+30.
 30   CONTINUE
C
      Y=ABS(X)
      M=ABS(N)
C
C *** IF REQUESTED ORDERS N IS .GT. NLIM PRACTICAL LIMIT, ONLY
C *** CALCULATE BESSEL FUNCTIONS UP TO ORDER = NLIM, REMAINDER SET = 0.0
C
      IF(M.GT.NLIM) M = NLIM
      FM=FLOAT(M)
C
      IF(Y.LT.FM)  GO TO 50
        IF(Y.GT.8.0) GO TO 40
          Q=Y+5.0*(Y**0.33333)
          GO TO 70
 40     CONTINUE
        Q=Y+10.0
        GO TO 70
 50   CONTINUE
      IF(Y.GT.8.0) GO TO 60
        Q=FM+5.0*(Y**0.33333)
        GO TO 70
 60   CONTINUE
```

```
      Q=FM+10.0
 70   CONTINUE
      NQ=Q
      IF(NQ.GT.500) NQ=500
C
      BESSEL(NQ+2)=0.0
      BESSEL(NQ+1)=1.0
C
      DO 80 I=1,NQ
        NN=NQ-I
        BESSEL(NN+1)=(2.0*FLOAT(NN+1)/Y)*BESSEL(NN+2)-BESSEL(NN+3)
 80   CONTINUE
C
      SUM=BESSEL(1)
      DO 90 I=3,NQ+2,2
        SUM=SUM+2.0*BESSEL(I)
 90   CONTINUE
C
      DO 100 I=1,N
        BJ(I)=BESSEL(I)/SUM
 100  CONTINUE
C
      RETURN
      END
```

Sample CRPFAN Input File:

Rotor 1 Geometry

+

Operating Conditions

```
$INPUT
DT=2.0538 HTR=0.4247 NB=11 NCASE=1
XPCA=.2394 DIST=4.5 NTH=17 NSL=1 NHM=10
VO=225.8 RPM=7633 SHP=229.82 PAMB=13.753 TAMB=71.70
INST=1 BETA34=36.4
THAD=
 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170
NZ=10
Z=
1. .979 .9576 .9144 .8221 .7184 .594 .5189 .4762 .4279
ZMC=
.05978 .05538 .05012 .04047 .01881 -.00106 -.01271 -.01086
-.00674 .00092
YMC=
.06248 .05609 .04970 .03770 .01654 .00035 -.00568 -.00341
-.00035 .00248
CHORD=
.0327 .0561 .0703 .088 .1058 .1200 .1278 .1278 .1271 .126
TMOC=
.04 .025 .0218 .02 .023 .0299 .0462 .064 .0805 .1025
BETAP=
43.4 46.2 48 50.2 53.8 58.3 64.8 68.9 70.4 70.7
THETA=
.0383 .056 .072 .097 .145 .202 .272 .3325 .381 .44
ITDO=5  NALE=10 NBTE=10 ALE=10*0.  BTE=10*0.  XTM=-0.1
ILDO=3 XLM=-.25 INF=0
IDS=0  IPTO=1
ISHAPE=2 BETAW=0.0 NWHM=15 VREF=10.0 NSTEP=101
IPRNTW=0   SCD=10*.02   WTIV=0.0
ITPVTX=1 IHBVTX=0 SBN=.5 0 TAU=10. CI=2. TVTI=2.
$
```

Sample CRPFAN Input File:

Rotor 2 Geometry

```
$INPUT
DT=1.9952 HTR=0.4148  NB=9
NZ=10 SHP=239.35 RPM=7695 BETA34=36.5
Z=
1. .9776 .9554 .9105 .8015 .7105 .5845 .5078 .4644 .4159
ZMC=
.05442 .04931 .04565 .03616 .01351 -.00314 -.01665 -.01454
-.00877 -.00073
YMC=
.06574 .05917 .05340 .04127 .01812 .00365 -.00533 -.00373
-.00095 .00329
CHORD=
.03433 .05771 .07378 .09167 .11103 .12345 .13149 .13141
.130 .12856
TMOC=
.04 .025 .0212 .021 .0238 .0298 .0471 .0662 .086 .1048
BETAP=
45.3 47.7 48.7 50.3 53.3 56 60.6 63.7 64.8 64.2
$
```

Sample CRPFAN Input File:

Installation Input

```
$INSTAL
NPCLCD=11,
NDES=12,
NRADP=10,
THETDP=5.,
CDP=0.02,
IPYLON=0,
ALD=
    1.850,  2.333,  2.418,  2.907,  3.292,  3.544,  3.559,  3.962,
 3.966,  4.188,  4.209,
CL=
    0.3571,  0.4099,  0.4192,  0.4727,  0.5147,  0.5423,  0.5439,  0.5880,
 0.5884,  0.6128,  0.6150,
CD=
    0.0035,  0.0056,  0.0059,  0.0074,  0.0096,  0.0119,  0.0120,  0.0161,
 0.0162,  0.0184,  0.0186,
ALDA=
    3.701,  4.022,  4.119,  4.444,  4.757,  4.916,  4.952,  5.203,
 5.254,  5.370,  5.404,
CLA=
    0.3036,  0.3329,  0.3417,  0.3714,  0.3999,  0.4144,  0.4177,  0.4406,
 0.4452,  0.4558,  0.4588,
CDA=
    0.0331,  0.0361,  0.0370,  0.0401,  0.0430,  0.0445,  0.0449,  0.0472,
 0.0477,  0.0488,  0.0491,
RADP=
  .4545, .4757, .5198, .5957, .7199, .8234, .9153, .9584,
  .9793, 1.0000,
XLEP=
  .7863, .7920, .8042, .8245, .8586, .8866, .9113, .9231,
  .9288, .9344,
CAMPD=
  10*0.0
XTEP=
  .1197, .1225, .1286, .1388, .1554, .1692, .1814, .1871,
  .1899, .1927,
PSPANV=
  7.50, 17.00, 26.00, 37.00, 46.00, 57.00, 67.00, 77.00,
  87.00, 92.00, 96.00, 98.00,
THFDG=.25,
CLACC=0.0,
ARAC=10.,
AATC=8.,
WSS=6.8,
NPGC=32,
XWL=-1.02,
YWL=3.4,
ZWL=0.00,
ARAD=2.,
NHMAX=1,
NMODES=5,
NROT=-1,
FLEN=13.6,
FDIA=0.0,
FRAWC=.4,
NPHLAG=0,
NANG=1,
ARBDU=0.00,
```

```
$
```

Sample CRPFAN Output File:

"Isolated" Prediction

```
COUNTER-ROTATING PROPELLER NOISE PREDICTION PROGRAM

BETAP SURFACE APPROXIMATION MODEL
WAKE MODEL INCLUDED

   *** ROTOR 1 INPUT PARAMETERS ***

      PROPELLER TIP DIAMETER DT =   2.0538  FT.
      NUMBER OF BLADES NB       =    11
      SHAFT HORSEPOWER SHP      =     229.82
      ROTATIVE SPEED (RPM)      =     7633.0
      FLIGHT VELOCITY VO (FPS)  =   225.80
      AMBIENT TEMPERATURE TAMB  =    71.70  DEG. F
      AMBIENT PRESSURE PAMB     =   13.753 PSIA
      NUMBER OF PROPELLERS NP   =   1

   *** ROTOR 2 INPUT PARAMETERS ***

      PROPELLER TIP DIAMETER DT =   1.9952  FT.
      NUMBER OF BLADES NB       =     9
      SHAFT HORSEPOWER SHP      =     239.35
      ROTATIVE SPEED (RPM)      =     7695.0
      FLIGHT VELOCITY VO (FPS)  =   225.80
      AMBIENT TEMPERATURE TAMB  =    71.70  DEG. F
      AMBIENT PRESSURE PAMB     =   13.753 PSIA
      NUMBER OF PROPELLERS NP   =   1


      XPCA = 0.2394
      IRW  = 1
      IWAKE= 2
      IDS  = 0
      INF  = 0
      INFS = 1


      TIP SPEED MACH NO. =  0.7263
      FLIGHT    MACH NO. =  0.1998
      HELICAL   MACH NO. =  0.7533


      TIP SPEED MACH NO. =  0.7114
      FLIGHT    MACH NO. =  0.1998
      HELICAL   MACH NO. =  0.7389
```

TABLE OF BLADE SECTION PROPERTIES - ROTOR 1

SECTION	R/RT	CHORD	TM/C	BETAP	MCA	FA	CL	CD
1	1.0000	0.0327	0.0400	22.95	0.0865	0.0005	1.0191	0.0200
2	0.9790	0.0561	0.0250	25.75	0.0788	-0.0022	0.8490	0.0200
3	0.9576	0.0703	0.0218	27.55	0.0705	-0.0034	0.8506	0.0200
4	0.9144	0.0880	0.0200	29.75	0.0552	-0.0031	0.8710	0.0200
5	0.8221	0.1058	0.0230	33.35	0.0249	-0.0022	0.9642	0.0200
6	0.7184	0.1200	0.0299	37.85	-0.0007	-0.0009	1.0191	0.0200
7	0.5940	0.1278	0.0462	44.35	-0.0139	-0.0003	1.0352	0.0200
8	0.5189	0.1278	0.0640	48.45	-0.0114	-0.0007	1.0764	0.0200
9	0.4762	0.1271	0.0805	49.95	-0.0065	-0.0019	1.1154	0.0200
10	0.4279	0.1260	0.1025	50.25	0.0017	-0.0020	1.1343	0.0200

TABLE OF BLADE SECTION PROPERTIES - ROTOR 2

SECTION	R/RT	CHORD	TM/C	BETAP	MCA	FA	CL
1	1.0000	0.0343	0.0400	27.02	0.0849	-0.0084	1.3232
2	0.9776	0.0577	0.0250	29.42	0.0763	-0.0106	1.0958
3	0.9554	0.0738	0.0212	30.42	0.0695	-0.0100	1.0520
4	0.9105	0.0917	0.0210	32.02	0.0542	-0.0087	1.0479
5	0.8015	0.1110	0.0238	35.02	0.0217	-0.0065	1.0582
6	0.7105	0.1235	0.0298	37.72	-0.0006	-0.0048	1.0918
7	0.5845	0.1315	0.0471	42.32	-0.0171	-0.0035	1.0337
8	0.5078	0.1314	0.0662	45.42	-0.0147	-0.0031	1.0181
9	0.4644	0.1300	0.0860	46.52	-0.0083	-0.0029	1.0196
10	0.4159	0.1286	0.1048	45.92	0.0008	-0.0033	0.9983

ROTOR 1

```
EFFECTIVE AERO SHP   =  229.82
ROTOR ANNULUS AREA   =    2.715 SQ FT
DISK LOADING SHP/D2 =   54.48
DISK LOADING SHP/AA =   84.64
POWER COEFFICIENT CP=    0.774
BLADE TIP SPEED UT   =  820.83 FT/SEC
BLADE PITCH ANGLE    =   36.40 DEG.
```

ROTOR 2

```
EFFECTIVE AERO SHP   =  239.35
ROTOR ANNULUS AREA   =    2.589 SQ FT
DISK LOADING SHP/D2 =   60.13
DISK LOADING SHP/AA =   92.46
POWER COEFFICIENT CP=    0.909
BLADE TIP SPEED UT   =  803.88 FT/SEC
BLADE PITCH ANGLE    =   36.50 DEG.
```

*** TONE HARMONIC SPL SPECTRA ***

4.50 FT. SIDELINE

THAO-R1		10.0	20.0	30.0	40.0	50.0	60.0	70.0	80.0	90.0	100.0	110.0	120.0	130.0	140.0	150.0	160.0	170.0
THAE-R1		8.0	16.1	24.3	32.6	41.2	50.0	59.2	68.7	78.5	88.7	99.2	110.0	121.2	132.6	144.3	156.1	168.0
THAO-R2		9.8	19.3	28.5	37.5	46.5	55.5	64.7	74.1	83.8	93.8	104.3	115.1	126.1	137.3	148.4	159.2	169.8
THAE-R2		7.9	15.5	23.0	30.6	38.2	46.0	54.3	63.0	72.3	82.3	93.1	104.7	116.8	129.5	142.3	155.2	167.8
M	FREQ.																	
1	1399.4	0.0	9.4	46.7	69.8	84.5	95.4	105.1	112.4	116.5	117.4	115.2	109.3	99.2	83.3	59.1	21.1	0.0
2	2798.8	0.0	0.0	0.0	17.9	47.5	67.0	81.3	91.7	97.1	97.3	92.0	80.4	60.7	30.3	0.0	0.0	0.0
3	4198.1	0.0	0.0	0.0	0.0	11.1	40.6	60.4	73.3	79.5	78.7	70.3	52.7	23.5	0.0	0.0	0.0	0.0
4	5597.5	0.0	0.0	0.0	0.0	0.0	13.8	39.4	55.1	62.2	60.4	48.8	25.4	0.0	0.0	0.0	0.0	0.0
5	6996.9	0.0	0.0	0.0	0.0	0.0	0.0	18.7	37.5	45.4	42.5	27.7	0.0	0.0	0.0	0.0	0.0	0.0
6	8396.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	19.4	28.2	24.4	6.6	0.0	0.0	0.0	0.0	0.0	0.0
7	9795.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.2	11.7	6.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	11195.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	12594.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	13993.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	1154.3	0.0	23.5	53.3	72.0	85.7	97.9	108.2	115.8	120.8	123.2	122.8	119.2	111.5	98.4	77.9	45.1	0.0
12	2308.5	0.0	0.0	0.0	24.4	48.8	67.0	82.7	94.6	102.1	105.2	103.5	96.0	81.3	56.5	17.6	0.0	0.0
13	3462.8	0.0	0.0	0.0	0.0	14.0	40.4	60.9	76.1	85.6	89.0	85.8	74.4	52.5	16.1	0.0	0.0	0.0
14	4617.0	0.0	0.0	0.0	0.0	0.0	13.8	39.4	57.9	69.2	72.9	68.2	53.1	24.1	0.0	0.0	0.0	0.0
15	5771.3	0.0	0.0	0.0	0.0	0.0	0.0	18.6	40.3	53.4	57.4	51.2	32.1	0.0	0.0	0.0	0.0	0.0
16	6925.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	23.3	38.1	42.3	34.5	11.6	0.0	0.0	0.0	0.0	0.0
17	8079.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.9	22.6	27.1	17.8	0.0	0.0	0.0	0.0	0.0	0.0
18	9234.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.4	12.0	1.1	0.0	0.0	0.0	0.0	0.0	0.0
19	10388.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	11542.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
21	2553.6	105.7	103.5	117.3	115.2	112.1	114.9	107.9	104.2	104.1	102.2	98.0	94.6	99.7	101.5	107.0	104.1	104.2
22	3707.9	64.0	102.0	112.9	97.7	110.5	107.8	104.5	101.8	104.9	106.0	109.2	106.9	111.0	116.3	112.9	109.0	71.4
23	4862.1	0.0	33.1	78.6	102.4	111.3	103.2	106.1	105.3	102.4	101.7	108.0	111.1	114.7	116.5	98.0	54.0	0.0
24	6016.4	0.0	0.0	31.1	71.5	94.1	104.5	104.2	94.3	98.6	99.6	109.2	118.4	113.7	93.5	53.1	0.0	0.0
25	7170.6	0.0	0.0	0.0	37.2	72.2	90.0	91.7	102.6	112.2	116.4	117.5	111.7	90.9	54.8	4.5	0.0	0.0
26	8324.9	0.0	0.0	0.0	0.0	44.4	71.9	78.9	97.5	108.7	112.5	110.0	97.7	73.4	33.9	0.0	0.0	0.0
27	9479.1	0.0	0.0	0.0	0.0	10.8	48.0	50.3	89.1	101.8	105.6	101.5	87.5	58.7	4.4	0.0	0.0	0.0
28	10633.4	0.0	0.0	0.0	0.0	0.0	20.0	48.7	79.3	93.4	97.4	92.0	74.4	37.9	0.0	0.0	0.0	0.0
29	11787.6	0.0	0.0	0.0	0.0	0.0	4.3	37.0	67.4	83.1	87.3	80.1	58.1	14.0	0.0	0.0	0.0	0.0
30	12941.9	0.0	0.0	0.0	0.0	0.0	0.0	21.1	53.9	71.5	75.6	65.9	38.7	0.0	0.0	0.0	0.0	0.0
31	3953.0	0.0	49.4	86.4	105.8	113.7	111.6	108.3	112.6	111.0	107.6	103.2	92.0	80.6	91.0	78.6	43.0	0.0
32	5107.3	108.5	105.0	110.6	107.9	94.4	99.9	99.5	106.1	104.7	99.2	83.4	90.6	92.8	96.7	99.9	101.8	97.8
33	6261.5	108.5	109.1	103.8	96.7	99.8	102.1	99.0	101.4	81.8	86.4	94.4	84.4	95.8	108.7	109.2	106.5	101.5
34	7415.8	33.4	100.8	109.8	102.1	100.4	102.4	97.9	99.1	90.3	97.3	98.7	101.6	102.5	109.5	110.0	103.3	33.4
35	8570.0	0.0	49.3	100.8	109.8	112.2	108.1	104.9	98.7	95.2	92.5	105.9	104.7	109.1	110.2	110.0	58.1	0.0
36	9724.3	0.0	0.0	65.2	104.3	108.4	111.0	105.0	88.7	100.8	108.0	105.6	112.6	100.2	116.8	83.0	0.0	0.0
37	10878.5	0.0	0.0	19.5	80.0	107.7	104.3	105.2	95.9	104.6	106.5	103.8	116.1	118.4	103.1	46.8	0.0	0.0
38	12032.8	0.0	0.0	0.0	48.0	91.4	108.3	96.1	97.5	107.3	112.1	118.1	110.3	115.0	81.8	4.6	0.0	0.0
39	13187.0	0.0	0.0	0.0	11.6	68.6	97.6	102.2	93.6	108.1	113.7	90.0	118.4	103.9	55.5	0.0	0.0	0.0
40	14341.3	0.0	0.0	0.0	0.0	41.3	80.4	93.9	101.5	108.1	110.7	117.6	114.4	88.4	25.6	0.0	0.0	0.0
41	5352.4	0.0	0.0	15.1	58.0	84.0	98.7	105.6	107.1	105.8	103.7	100.8	93.8	76.8	36.0	6.3	0.0	0.0
42	6506.6	0.0	75.5	103.7	99.2	101.8	91.6	100.5	102.0	103.7	99.8	92.2	87.2	85.4	90.2	95.6	71.5	0.0
43	7660.9	98.5	90.4	106.3	98.7	94.4	98.2	103.1	96.2	94.3	91.2	84.7	85.0	92.1	87.9	78.3	82.8	95.6
44	8815.2	98.7	101.4	96.4	108.1	100.0	104.5	99.6	92.6	86.6	79.8	88.4	82.9	96.1	97.0	95.3	92.4	91.6
45	9969.4	75.6	85.9	94.8	98.1	101.8	96.8	98.9	92.9	79.5	86.6	94.5	103.1	99.7	101.0	94.8	98.3	77.1
46	11123.7	0.0	90.8	94.4	99.8	98.7	96.8	97.0	92.8	89.0	94.6	90.6	97.8	102.4	99.6	105.5	96.9	0.0

47	12277.9	0.0	43.1	101.4	101.5	92.0	100.4	86.0	95.0	96.3	99.2	98.4	100.3	106.8	103.2	106.5	58.3	0.0
48	13432.2	0.0	0.0	73.7	106.0	94.7	99.2	97.8	92.6	84.9	90.0	99.4	94.3	99.5	103.9	93.9	4.2	0.0
49	14586.4	0.0	0.0	34.2	93.8	97.0	97.6	96.3	92.8	96.8	96.6	105.6	103.0	93.6	109.0	65.7	0.0	0.0
50	15740.7	0.0	0.0	0.0	70.8	103.3	100.0	94.7	89.1	90.5	94.4	105.2	104.0	102.4	100.4	29.8	0.0	0.0
51	6751.8	0.0	0.0	0.0	4.0	46.0	72.5	88.1	95.9	98.3	96.9	91.2	78.1	52.0	3.4	0.0	0.0	0.0
52	7906.0	0.0	0.0	62.5	96.0	102.9	104.3	90.1	103.9	96.5	93.9	95.7	79.9	74.6	85.2	62.2	0.0	0.0
53	9060.3	10.6	90.8	93.5	104.0	99.0	94.3	98.5	82.3	93.2	91.1	74.3	80.8	77.4	86.4	93.2	87.5	7.3
54	10214.5	98.6	94.9	96.3	98.8	99.9	98.9	95.9	98.7	82.5	70.2	69.1	81.7	85.0	91.9	91.6	94.0	92.2
55	11368.8	94.4	93.2	95.9	81.3	100.9	80.2	98.1	95.9	85.2	80.6	70.3	80.4	92.9	90.4	97.6	93.7	85.2
56	12523.0	97.1	99.5	100.0	86.0	98.8	101.3	82.5	90.9	83.8	71.4	83.6	89.9	85.6	96.1	99.3	94.4	94.4
57	13677.3	43.5	95.6	93.5	98.7	86.6	93.3	93.3	79.0	81.5	77.4	90.3	90.2	95.4	87.1	94.6	93.2	48.1
58	14831.5	0.0	85.2	103.1	100.1	86.0	89.4	87.9	88.9	75.8	84.0	87.6	85.3	95.4	83.1	97.5	93.9	0.0
59	15985.8	0.0	42.3	101.8	96.3	97.4	94.6	95.7	84.0	88.8	94.0	96.8	96.8	96.4	100.5	96.2	58.7	0.0
60	17140.0	0.0	0.0	84.0	89.2	91.0	87.8	73.1	92.1	89.0	82.4	92.3	93.6	97.6	97.2	101.2	9.0	0.0
61	8151.2	0.0	0.0	0.0	0.0	4.1	42.2	66.1	79.6	85.2	84.1	75.8	56.9	21.5	0.0	0.0	0.0	0.0
62	9305.4	0.0	0.0	1.4	61.1	93.2	104.8	86.7	101.5	96.0	92.2	94.3	68.0	79.2	56.1	7.4	0.0	0.0
63	10459.7	0.0	27.1	89.8	101.1	76.4	98.6	97.8	96.2	95.4	91.3	86.8	82.2	65.0	87.3	87.5	30.0	0.0
64	11613.9	20.7	98.9	95.8	94.4	97.5	99.5	83.4	86.5	94.8	87.3	72.5	71.7	83.2	60.8	92.2	93.7	18.9
65	12768.2	96.7	95.2	99.1	97.9	99.7	95.5	98.5	94.8	91.9	89.4	72.0	82.7	88.6	90.9	93.5	90.6	92.3
66	13922.4	94.0	91.4	99.1	98.6	93.9	97.4	96.2	93.5	92.2	84.8	67.0	87.6	81.2	77.7	94.6	91.5	88.6
67	15076.7	91.6	96.9	90.4	98.3	95.5	96.0	95.6	87.7	69.2	64.6	79.2	88.1	93.2	74.8	94.4	91.0	92.0
68	16230.9	78.5	91.5	98.3	97.0	97.2	93.4	95.3	84.3	83.9	72.6	77.3	93.2	91.2	98.2	96.7	95.5	81.3
69	17385.2	11.5	100.5	96.8	99.4	98.7	87.0	93.6	82.9	82.4	76.8	78.1	93.5	94.7	95.3	97.7	91.2	19.5
70	18539.4	0.0	81.2	98.8	98.8	98.7	92.8	95.1	82.0	73.5	84.3	91.1	94.6	97.0	98.4	100.0	92.8	0.0
71	9550.5	0.0	0.0	0.0	0.0	0.0	6.6	38.5	57.3	65.7	65.1	54.5	30.2	0.0	0.0	0.0	0.0	0.0
72	10704.8	0.0	0.0	0.0	12.5	62.5	90.0	100.7	98.1	78.4	55.0	90.1	89.1	66.5	3.7	0.0	0.0	0.0
73	11859.0	0.0	0.0	41.5	90.6	97.9	93.9	87.6	94.2	91.6	91.0	80.5	68.5	55.5	82.3	48.9	0.0	0.0
74	13013.3	0.0	48.4	99.2	92.1	96.5	96.2	85.6	73.2	92.1	88.1	78.1	65.2	69.6	74.7	85.6	52.2	0.0
75	14167.5	25.3	97.9	93.0	95.4	92.4	78.6	93.0	81.3	84.9	82.2	78.6	70.6	83.7	82.0	87.5	85.9	24.8
76	15321.8	93.8	93.4	95.4	90.2	94.6	92.1	87.7	90.8	88.8	82.9	64.6	77.6	82.9	89.0	84.8	83.0	89.6
77	16476.1	90.7	94.7	95.8	72.6	90.3	84.0	83.7	90.2	87.4	80.0	55.5	78.2	87.7	80.2	87.7	87.3	88.1
78	17630.3	91.6	88.4	77.3	93.4	85.8	77.7	92.9	79.3	72.7	71.8	76.0	87.1	90.1	87.7	90.8	91.7	88.6
79	18784.6	93.3	85.5	74.2	84.9	95.2	94.6	80.3	87.5	82.0	64.4	68.4	70.8	83.4	73.1	92.7	93.2	92.8
80	19938.8	50.7	92.9	95.3	95.1	70.0	86.2	90.8	89.2	82.2	63.0	78.8	85.1	92.1	94.3	83.4	92.4	55.8
81	10949.9	0.0	0.0	0.0	0.0	0.0	0.0	7.3	31.1	42.3	42.1	29.2	0.0	0.0	0.0	0.0	0.0	0.0
82	12104.2	0.0	0.0	0.0	0.0	23.2	63.6	85.2	93.8	94.4	92.5	88.6	75.0	40.5	0.0	0.0	0.0	0.0
83	13258.4	0.0	0.0	0.0	53.9	90.7	89.9	90.7	92.4	73.0	78.8	83.2	75.8	67.1	56.4	0.0	0.0	0.0
84	14412.7	0.0	0.0	66.1	95.6	77.1	91.2	86.5	89.1	61.8	70.8	81.8	68.5	66.7	77.8	71.2	0.0	0.0
85	15566.9	0.0	61.6	83.7	85.0	92.9	90.0	90.6	78.0	83.9	79.2	71.4	59.3	73.7	80.7	82.7	65.0	0.0
86	16721.2	26.8	84.2	88.0	80.9	84.9	90.6	89.6	80.3	66.5	74.8	72.3	70.0	57.1	81.6	75.8	82.5	26.8
87	17875.4	89.0	84.5	88.2	91.6	86.9	89.4	75.7	84.3	82.1	73.3	54.1	64.0	76.6	83.6	70.6	84.3	84.5
88	19029.7	83.1	56.3	90.8	90.5	87.5	89.3	77.6	83.2	78.4	60.9	35.7	60.3	76.6	82.4	65.6	82.4	82.9
89	20183.9	73.4	76.7	82.6	86.1	76.4	89.4	78.1	73.7	77.3	72.2	69.7	74.4	79.0	86.5	86.5	79.4	75.7
90	21338.2	80.3	81.5	87.8	86.6	87.5	87.5	88.1	84.9	76.8	56.4	65.2	81.8	78.7	87.4	86.4	86.5	83.6
91	12349.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.6	20.5	20.7	5.6	0.0	0.0	0.0	0.0	0.0	0.0
92	13503.6	0.0	0.0	0.0	0.0	0.0	35.2	66.3	81.9	87.2	86.0	78.0	56.9	11.9	0.0	0.0	0.0	0.0
93	14657.8	0.0	0.0	0.0	10.7	67.5	92.7	82.3	90.2	78.5	79.1	83.1	62.0	67.8	18.9	0.0	0.0	0.0
94	15812.1	0.0	0.0	17.1	81.7	84.2	91.1	87.5	81.2	87.7	84.1	64.8	69.5	47.8	75.5	31.6	0.0	0.0
95	16966.3	0.0	0.0	83.9	92.6	90.2	86.4	88.8	67.1	78.5	68.3	75.3	63.3	69.0	75.6	81.5	7.1	0.0
96	18120.6	0.0	73.8	90.3	90.2	89.2	74.8	85.8	78.4	79.7	66.6	62.2	50.4	74.2	74.2	81.4	75.0	0.0
97	19274.8	30.7	87.7	85.6	89.1	78.3	80.1	87.1	79.2	79.3	78.2	70.0	66.7	76.7	80.4	66.3	78.7	30.3
98	20429.1	87.7	86.5	82.9	73.7	81.2	82.4	81.7	79.8	66.8	66.5	55.8	58.5	78.5	80.7	81.1	61.9	82.0
99	21583.3	71.7	86.9	89.2	81.7	86.4	82.1	84.6	78.9	60.0	68.3	43.7	72.2	77.7	82.4	78.5	80.6	79.3
100	22737.6	84.4	86.3	84.7	86.7	75.3	82.7	80.0	71.8	79.5	72.6	65.7	67.4	77.6	81.5	84.7	80.7	78.2
101	13748.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
102	14903.0	0.0	0.0	0.0	0.0	0.0	5.7	45.2	66.8	75.6	74.9	63.8	36.3	0.0	0.0	0.0	0.0	0.0
103	16057.2	0.0	0.0	0.0	0.0	38.3	78.6	93.3	79.6	87.5	86.2	59.5	81.6	53.5	0.0	0.0	0.0	0.0
104	17211.5	0.0	0.0	0.0	51.7	91.5	90.9	89.8	87.6	86.5	85.2	76.1	75.1	52.5	58.1	0.0	0.0	0.0

105	18365.7	0.0	0.0	48.1	93.2	85.0	84.6	74.4	88.0	83.6	81.6	77.5	54.1	55.7	74.3	58.9	0.0	0.0
106	19520.0	0.0	20.0	92.1	89.1	89.1	63.6	84.9	78.4	78.4	63.2	67.7	56.5	70.5	73.2	58.2	29.1	0.0
107	20674.2	0.0	82.7	88.8	70.9	88.0	87.8	71.2	80.1	72.2	61.7	63.4	34.1	63.6	63.0	78.8	81.3	0.0
108	21828.5	34.3	85.6	82.4	61.0	75.9	86.6	80.8	80.8	81.6	77.7	68.8	63.5	72.3	77.9	79.5	79.4	34.4
109	22982.7	86.4	80.6	77.6	87.0	70.4	86.2	85.5	75.4	75.8	75.2	56.9	68.7	61.2	75.8	82.1	80.0	81.8
110	24137.0	54.1	76.5	86.5	85.3	86.7	85.6	86.4	76.7	76.2	74.0	40.4	73.8	79.0	81.6	82.2	75.0	75.0
111	15148.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
112	16302.3	0.0	0.0	0.0	0.0	0.0	0.0	20.0	47.1	59.0	58.6	44.8	11.3	0.0	0.0	0.0	0.0	0.0
113	17456.6	0.0	0.0	0.0	0.0	2.5	56.0	82.6	89.8	84.4	81.1	83.6	71.9	30.0	0.0	0.0	0.0	0.0
114	18610.8	0.0	0.0	0.0	11.1	72.6	90.3	87.2	84.9	82.8	81.8	74.0	72.8	66.2	25.6	0.0	0.0	0.0
115	19765.1	0.0	0.0	0.0	76.3	75.2	79.0	85.5	70.0	74.2	57.5	73.8	68.9	41.1	71.6	17.3	0.0	0.0
116	20919.3	0.0	0.0	67.8	84.7	85.7	82.2	85.1	76.9	73.8	75.6	73.8	64.7	59.6	72.0	73.2	0.0	0.0
117	22073.6	0.0	35.8	75.8	75.7	81.6	82.7	77.0	78.9	71.0	53.6	51.7	41.6	65.8	68.8	69.7	44.4	0.0
118	23227.8	0.0	85.4	85.6	85.5	84.8	79.1	81.9	76.0	66.2	72.1	61.3	53.1	68.0	72.7	72.9	79.5	0.0
119	24382.1	35.0	79.2	77.6	84.8	61.5	79.4	65.8	79.2	79.0	72.5	64.8	45.2	67.0	72.0	77.9	70.2	35.6
120	25536.3	83.7	82.8	64.7	82.3	61.2	77.3	83.1	59.5	77.5	73.1	56.5	68.4	74.7	66.0	71.7	75.2	78.4
121	16547.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
122	17701.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	24.7	39.5	39.4	23.1	0.0	0.0	0.0	0.0	0.0	0.0
123	18856.0	0.0	0.0	0.0	0.0	0.0	28.3	64.8	80.8	84.0	82.4	76.3	54.8	1.3	0.0	0.0	0.0	0.0
124	20010.2	0.0	0.0	0.0	0.0	44.3	82.0	72.6	75.7	80.8	78.3	63.1	53.7	56.2	0.0	0.0	0.0	0.0
125	21164.5	0.0	0.0	0.0	44.1	86.1	78.6	83.1	76.8	78.2	74.2	55.6	66.5	54.2	54.4	0.0	0.0	0.0
126	22318.7	0.0	0.0	24.5	85.5	81.3	80.4	71.8	79.1	77.5	73.0	66.7	45.6	51.1	66.8	42.2	0.0	0.0
127	23473.0	0.0	0.0	78.5	80.3	81.4	81.7	64.6	65.6	65.0	70.5	65.4	41.7	55.3	45.6	73.0	0.0	0.0
128	24627.2	0.0	47.2	82.0	79.4	70.9	81.3	79.6	76.1	59.6	62.4	43.0	40.5	44.3	70.2	69.1	54.6	0.0
129	25781.5	0.0	83.3	81.5	67.2	80.1	77.3	79.3	68.0	72.7	70.5	56.5	54.8	67.1	71.7	57.1	57.9	0.0
130	26935.7	34.2	79.2	71.7	74.7	68.9	78.4	75.6	76.8	72.1	56.2	57.9	54.7	69.3	56.8	69.8	73.7	35.1
131	17946.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
132	19101.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.4	21.1	21.4	2.8	0.0	0.0	0.0	0.0	0.0	0.0
133	20255.3	0.0	0.0	0.0	0.0	0.0	0.2	46.0	68.9	76.9	75.9	65.6	36.7	0.0	0.0	0.0	0.0	0.0
134	21409.6	0.0	0.0	0.0	0.0	13.4	66.5	84.6	75.5	77.2	73.3	73.3	72.9	39.8	0.0	0.0	0.0	0.0
135	22563.8	0.0	0.0	0.0	6.8	72.9	58.8	80.7	76.7	78.1	74.9	62.0	67.9	55.0	27.6	0.0	0.0	0.0
136	23718.1	0.0	0.0	0.0	68.6	79.9	80.4	80.8	74.8	72.6	73.5	69.8	63.5	22.8	65.6	1.8	0.0	0.0
137	24872.3	0.0	0.0	47.8	59.3	78.6	78.8	77.0	74.0	75.2	73.5	69.5	59.1	50.0	65.9	60.6	0.0	0.0
138	26026.6	0.0	0.0	83.3	81.6	66.6	79.0	78.8	72.5	64.1	68.4	52.6	53.0	38.8	67.3	69.3	5.6	0.0
139	27180.8	0.0	58.2	79.9	79.2	56.8	63.2	74.2	75.3	63.4	68.6	44.3	37.0	61.9	63.5	66.4	63.6	0.0
140	28335.1	0.0	75.4	77.2	80.0	78.5	77.7	73.8	65.6	73.4	70.3	59.0	53.4	51.0	68.7	60.9	71.9	0.0
141	19346.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
142	20500.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	2.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0
143	21654.7	0.0	0.0	0.0	0.0	0.0	0.0	25.3	54.5	66.2	65.7	52.2	16.5	0.0	0.0	0.0	0.0	0.0
144	22809.0	0.0	0.0	0.0	0.0	0.0	46.5	77.8	81.4	69.9	71.5	73.2	67.0	18.8	0.0	0.0	0.0	0.0
145	23963.2	0.0	0.0	0.0	0.0	51.4	83.7	80.6	72.8	77.0	73.5	45.1	67.0	58.6	0.0	0.0	0.0	0.0
146	25117.5	0.0	0.0	0.0	40.6	83.6	79.5	77.8	55.8	56.7	60.3	65.8	64.2	24.9	54.2	0.0	0.0	0.0
147	26271.7	0.0	0.0	5.4	81.8	80.8	75.1	75.1	74.2	64.5	53.7	44.6	55.3	45.6	63.7	29.9	0.0	0.0
148	27426.0	0.0	0.0	65.4	61.5	78.8	75.1	74.8	41.7	72.7	71.7	65.1	50.9	51.4	56.4	70.1	0.0	0.0
149	28580.2	0.0	8.3	65.8	77.4	69.2	71.8	71.8	74.7	63.1	67.7	37.7	42.9	54.6	58.6	59.3	23.0	0.0
150	29734.5	0.0	66.9	74.6	67.7	63.8	76.1	58.5	74.0	68.4	68.5	36.9	27.3	63.0	53.2	62.6	69.6	0.0
151	20745.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
152	21899.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
153	23054.1	0.0	0.0	0.0	0.0	0.0	0.0	0.9	36.0	51.1	51.0	34.6	0.0	0.0	0.0	0.0	0.0	0.0
154	24208.4	0.0	0.0	0.0	0.0	0.0	21.2	63.2	78.6	77.1	74.3	73.4	53.1	0.0	0.0	0.0	0.0	0.0
155	25362.6	0.0	0.0	0.0	0.0	22.7	73.2	63.4	66.0	71.1	61.2	63.6	41.6	46.3	0.0	0.0	0.0	0.0
156	26516.9	0.0	0.0	0.0	3.7	73.3	78.4	74.2	66.9	66.2	56.7	58.0	61.7	41.2	29.0	0.0	0.0	0.0
157	27671.1	0.0	0.0	0.0	62.7	78.9	61.1	75.5	74.6	73.7	69.2	65.1	60.7	35.2	61.2	0.0	0.0	0.0
158	28825.4	0.0	0.0	28.9	70.3	66.8	69.4	64.6	73.1	70.6	63.4	62.9	45.6	49.7	61.5	48.6	0.0	0.0
159	29979.6	0.0	0.0	74.6	63.6	60.2	62.9	72.8	68.0	68.1	66.3	55.3	50.3	52.5	61.8	60.5	0.0	0.0
160	31133.9	0.0	21.9	75.5	49.8	70.9	67.8	68.0	73.1	63.3	66.5	40.4	18.3	57.6	60.8	43.0	35.3	0.0
161	22145.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
162	23299.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

163	24453.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	15.4	33.5	33.8	14.7	0.0	0.0	0.0	0.0	0.0	0.0
164	25607.8	0.0	0.0	0.0	0.0	0.0	0.0	44.6	68.0	73.9	72.4	64.3	35.0	0.0	0.0	0.0	0.0	0.0
165	26762.0	0.0	0.0	0.0	0.0	0.0	55.3	76.6	73.1	63.3	65.2	67.4	64.8	27.0	0.0	0.0	0.0	0.0
166	27916.3	0.0	0.0	0.0	0.0	52.5	73.4	70.9	62.1	63.3	47.3	54.2	58.7	52.2	0.0	0.0	0.0	0.0
167	29070.5	0.0	0.0	0.0	32.8	75.6	68.0	68.0	57.6	67.1	67.5	63.6	55.4	34.4	49.4	0.0	0.0	0.0
168	30224.8	0.0	0.0	0.0	73.9	74.8	69.6	70.7	53.7	65.4	64.8	56.4	52.0	32.6	57.2	13.3	0.0	0.0
169	31379.0	0.0	0.0	46.2	71.1	70.9	70.0	71.1	60.0	68.8	63.2	58.2	51.1	49.4	52.6	59.1	0.0	0.0
170	32533.3	0.0	0.0	75.1	69.5	67.5	64.5	61.8	67.7	63.3	62.6	46.9	34.0	52.4	51.8	63.6	0.0	0.0

*** 1/3-OCTAVE SPL SPECTRA ***

4.50 FT. SIDELINE

O.ANG	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170
FREQ																	
50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
63	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
80	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
100	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
125	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
160	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
200	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
250	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
315	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
400	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
500	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
630	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
800	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1250	0.0	23.6	54.1	74.0	88.1	99.8	109.9	117.4	122.2	124.2	123.5	119.6	111.7	98.5	77.9	45.1	0.0
1600	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2500	105.6	103.4	117.2	115.2	112.0	114.9	107.9	104.9	106.7	107.4	104.8	98.4	99.8	101.5	107.0	104.1	104.1
3150	0.0	0.0	0.0	0.0	14.0	40.4	60.8	76.1	85.6	89.0	85.7	74.4	52.5	16.1	0.0	0.0	0.0
4000	63.8	101.8	112.8	106.4	115.3	113.0	109.8	112.9	111.9	109.9	110.1	107.0	110.9	116.3	112.9	108.9	71.2
5000	108.2	104.9	110.5	108.9	111.3	105.8	109.3	111.0	109.2	106.6	108.7	111.1	114.7	116.5	102.0	101.6	97.6
6300	108.0	108.9	106.6	101.1	104.3	106.6	106.6	105.5	105.7	103.8	109.4	118.3	113.7	108.8	109.3	106.3	101.2
8000	101.0	104.0	111.7	112.6	113.1	111.5	108.3	108.4	113.9	117.9	118.4	112.9	110.1	112.9	112.9	103.5	96.6
10000	97.7	97.3	100.9	109.2	112.1	113.1	110.1	106.1	108.3	111.9	109.1	117.8	118.5	117.1	106.1	101.4	91.7
12500	100.4	103.6	107.2	108.7	107.1	111.0	106.8	104.7	111.1	116.0	118.0	118.9	115.8	107.0	108.2	100.4	96.5
16000	96.5	103.7	107.0	106.6	107.4	105.4	105.1	104.0	108.6	110.8	117.9	114.9	105.9	110.4	105.4	99.9	94.9
20000	91.6	95.1	101.5	101.9	101.4	99.9	99.0	95.6	92.6	90.0	91.6	94.9	98.1	99.9	100.5	97.2	91.9
OASPL	113.0	113.5	120.6	119.4	120.6	120.5	118.1	120.4	123.9	126.2	126.6	125.6	123.1	122.6	118.4	113.5	107.8
DBA	113.1	112.9	121.1	119.2	120.4	120.2	117.7	120.7	124.1	126.1	125.9	124.4	121.8	122.1	117.9	113.5	107.9
DBD	121.6	121.2	130.1	127.7	129.4	128.9	125.6	127.4	128.7	129.9	130.2	130.2	129.3	130.8	126.6	122.5	116.4

Sample CRPFAN Output File:

“Installed” Prediction

- C R P I E -

COUNTER ROTATION PROPELLER INSTALLATION EFFECTS PROGRAM

* * * I N P U T P A R A M E T E R S * * *

```
DENSITY (slug/cu.ft.)       =    0.002172
TEMP.(STATIC)- deg. F       =   71.70
PRESSURE(STATIC) psia       =   13.753
GAMMA                       =    1.400
FLIGHT MACH #               =    0.1998
AXIAL SEP. OF PCA (ins.)    =    5.900
ARBITRARY DELTA (u/U)       =    0.000
PROP. ANG. OF ATT. (deg.)   =    8.00

REPRESENTATIVE THICKNESS / MAX THICKNESS : 0.250
 (USED FOR THICKNESS NOISE CALCULATIONS)

        PYLON EFFECT NOT INCLUDED IN RESULTS
```

FRONT ROTOR

RPM	=	7633.0
TIP RADIUS (ins.)	=	12.323
NUMBER OF BLADES	=	11
70 % SPAN PITCH (deg.) (initial estimate)	=	33.135
POWER ABSORBED (HP)	=	229.8
AXIAL LOC. OF PCA (ins.)	=	0.00
PROP. ROTATION (FLA)	=	CLOCKWISE

CL and CD VERSUS ANGLE OF ATTACK (deg.)

ANGLE	CL	CD
1.85000	0.35710	0.00350
2.33300	0.40990	0.00560
2.41800	0.41920	0.00590
2.90700	0.47270	0.00740
3.29200	0.51470	0.00960
3.54400	0.54230	0.01190
3.55900	0.54390	0.01200
3.96200	0.58800	0.01610
3.96600	0.58840	0.01620
4.18800	0.61280	0.01840
4.20900	0.61500	0.01860

BLADE GEOMETRY

RAD(ins.)	CHORD(ins.)	BETAP(deg.)	YMC/DT	ZMC/DT	TMOC
5.273	3.105	70.70	0.0025	0.0009	0.1025
5.868	3.132	70.40	-0.0003	-0.0067	0.0805
6.394	3.150	68.90	-0.0034	-0.0109	0.0640
7.320	3.150	64.80	-0.0057	-0.0127	0.0462
8.853	2.957	58.30	0.0003	-0.0011	0.0299
10.131	2.608	53.80	0.0165	0.0188	0.0230
11.268	2.169	50.20	0.0377	0.0405	0.0200
11.800	1.733	48.00	0.0497	0.0501	0.0218
12.064	1.383	46.20	0.0561	0.0554	0.0250
12.323	0.806	43.40	0.0625	0.0598	0.0400

PER CENT SPAN LOCATIONS TO BE USED FOR ACOUSTIC CALCULATIONS

```
 7.50 %
17.00 %
26.00 %
37.00 %
46.00 %
57.00 %
67.00 %
77.00 %
87.00 %
92.00 %
96.00 %
98.00 %
```

REAR ROTOR

RPM	=	7695.0
TIP RADIUS (ins.)	=	11.971
NUMBER OF BLADES	=	9
70 % SPAN PITCH (deg.) (initial estimate)	=	34.383
POWER ABSORBED (HP)	=	239.4
AXIAL LOC. OF PCA (ins.)	=	5.90
PROP. ROTATION (FLA)	=	COUNTER-CLOCKWISE

CL and CD VERSUS ANGLE OF ATTACK (deg.)

ANGLE	CL	CD
3.70100	0.30360	0.03310
4.02200	0.33290	0.03610
4.11900	0.34170	0.03700
4.44400	0.37140	0.04010
4.75700	0.39990	0.04300
4.91600	0.41440	0.04450
4.95200	0.41770	0.04490
5.20300	0.44060	0.04720
5.25400	0.44520	0.04770
5.37000	0.45580	0.04880
5.40400	0.45880	0.04910

BLADE GEOMETRY

RAD(ins.)	CHORD(ins.)	BETAP(deg.)	YMC/DT	ZMC/DT	TMOC
4.979	3.078	64.20	0.0033	-0.0007	0.1048
5.559	3.113	64.80	-0.0010	-0.0088	0.0860
6.079	3.146	63.70	-0.0037	-0.0145	0.0662
6.997	3.148	60.60	-0.0053	-0.0167	0.0471
8.506	2.956	56.00	0.0037	-0.0031	0.0298
9.595	2.658	53.30	0.0181	0.0135	0.0238
10.900	2.195	50.30	0.0413	0.0362	0.0210
11.437	1.766	48.70	0.0534	0.0457	0.0212
11.703	1.382	47.70	0.0592	0.0493	0.0250
11.971	0.822	45.30	0.0657	0.0544	0.0400

```
AIRCRAFT WING PARAMETERS:         LIFT COEFFICIENT  =   0.000
                                  ASPECT RATIO      =  10.000
                                  SEMI-SPAN (ft.)   =   6.800

AIRCRAFT FUSELAGE PARAMETERS:     TOTAL LENGTH (ft.)     =  13.600
                                  MAX. DIAMETER (ft.)    =   0.000
                                  FRAC. AHEAD OF WING LL =   0.400

PROPELLER DISK CENTER LOCATION:   X DIST. DOWNSTREAM OF WING LL (ft.)     =  -1.020
                                  Y DIST. FROM A/C CL ALONG WING LL (ft.) =   3.400
                                  Z DIST. ABOVE WING LL (ft.)             =   0.000


                   *****END WRITE OF INPUT PARAMETERS*****
```

```
                  SOME OVERALL CALCULATED STEADY STATE QUANTITIES

COMPUTED 70 % SPAN PITCH (deg.): FRONT =    34.04895
                                  REAR =    35.12069

          TOTAL THRUSTS IN LBF: FRONT =    304.6866
                                  REAR =    280.1233

                TOTAL THRUST IN LBF =    584.8099

                 POWER (HP) : FRONT =    229.8200
                                  REAR =    239.3500

                  TOTAL POWER (HP) =    469.1700
```

** A C O U S T I C R E S U L T S **

FRONT ROTOR

PHASE LAG NOT INCLUDED IN RESULTS

HARMONIC # = 1 4.5 FT. ARC PREDICTIONS

OBSERVER ANGLES from INLET re: PROPELLER AXIS

	10.0	20.0	30.0	40.0	50.0	60.0	70.0	80.0	90.0	100.0	110.0	120.0	130.0	140.0	150.0	160.0	170.0
REF dB>	0.0	18.0	51.9	72.5	85.2	94.3	103.6	110.7	114.7	115.6	113.5	108.1	98.6	84.0	61.7	26.8	0.0
PHID	CRP INSTALLATION EFFECT DELTA dB re: Ref dB																
0.0	0.0	6.5	4.3	3.1	2.5	1.9	1.4	1.1	1.0	1.0	1.1	1.2	1.4	1.9	2.8	4.6	0.0
15.0	0.0	5.5	3.1	2.0	1.5	1.1	0.5	0.3	0.2	0.1	0.1	0.2	0.3	0.7	1.5	3.4	0.0
30.0	0.0	4.3	1.6	0.6	0.2	0.0	-0.5	-0.7	-0.8	-0.9	-1.0	-1.1	-1.1	-0.9	-0.2	1.9	0.0
45.0	0.0	2.7	-0.4	-1.4	-1.5	-1.4	-1.6	-1.8	-2.0	-2.1	-2.3	-2.5	-2.8	-2.9	-2.4	-0.1	0.0
60.0	0.0	0.9	-3.2	-4.1	-3.7	-3.0	-2.9	-3.0	-3.2	-3.4	-3.7	-4.2	-4.8	-5.5	-5.7	-2.8	0.0
75.0	0.0	-1.0	-7.5	-8.4	-6.5	-4.8	-4.2	-4.2	-4.3	-4.5	-5.0	-5.7	-6.9	-8.8	-11.0	-6.1	0.0
90.0	0.0	-2.3	-14.3	-15.7	-9.7	-6.4	-5.1	-4.8	-4.9	-5.1	-5.7	-6.6	-8.2	-11.5	-24.0	-8.6	0.0
105.0	0.0	-1.9	-11.7	-14.3	-10.2	-7.0	-5.2	-4.7	-4.7	-4.9	-5.4	-6.2	-7.5	-9.7	-12.9	-6.7	0.0
120.0	0.0	-0.2	-5.8	-7.6	-7.3	-5.9	-4.4	-3.9	-3.8	-4.0	-4.3	-4.7	-5.4	-6.3	-6.6	-3.4	0.0
135.0	0.0	1.7	-2.1	-3.7	-4.3	-4.1	-3.1	-2.7	-2.7	-2.7	-2.9	-3.1	-3.3	-3.4	-3.0	-0.6	0.0
150.0	0.0	3.4	0.4	-1.1	-2.0	-2.3	-1.7	-1.5	-1.4	-1.5	-1.5	-1.5	-1.5	-1.3	-0.6	1.5	0.0
165.0	0.0	4.8	2.2	0.8	-0.2	-0.8	-0.5	-0.4	-0.3	-0.3	-0.3	-0.2	0.0	0.4	1.2	3.1	0.0
180.0	0.0	6.0	3.5	2.2	1.2	0.5	0.5	0.6	0.6	0.7	0.7	0.9	1.2	1.7	2.5	4.4	0.0
195.0	0.0	6.9	4.6	3.2	2.3	1.5	1.4	1.4	1.4	1.5	1.6	1.8	2.1	2.7	3.6	5.4	0.0
210.0	0.0	7.6	5.3	4.0	3.1	2.3	2.0	2.0	2.0	2.1	2.3	2.5	2.9	3.5	4.4	6.2	0.0
225.0	0.0	8.1	5.9	4.6	3.7	2.9	2.6	2.5	2.5	2.6	2.8	3.0	3.4	4.0	5.0	6.8	0.0
240.0	0.0	8.5	6.3	5.0	4.2	3.3	2.9	2.8	2.8	2.9	3.1	3.4	3.8	4.5	5.4	7.2	0.0
255.0	0.0	8.7	6.6	5.3	4.4	3.6	3.2	3.0	3.0	3.1	3.3	3.6	4.1	4.7	5.7	7.4	0.0
270.0	0.0	8.8	6.7	5.5	4.6	3.8	3.3	3.1	3.1	3.2	3.4	3.7	4.2	4.8	5.8	7.5	0.0
285.0	0.0	8.8	6.7	5.5	4.6	3.8	3.3	3.1	3.1	3.2	3.4	3.7	4.1	4.7	5.7	7.4	0.0
300.0	0.0	8.6	6.5	5.3	4.5	3.7	3.1	2.9	2.9	3.0	3.2	3.5	3.9	4.5	5.5	7.2	0.0
315.0	0.0	8.3	6.2	5.0	4.3	3.5	2.9	2.7	2.6	2.7	2.9	3.1	3.5	4.1	5.1	6.9	0.0
330.0	0.0	7.9	5.8	4.6	3.9	3.1	2.5	2.3	2.2	2.3	2.4	2.7	3.0	3.6	4.5	6.3	0.0
345.0	0.0	7.3	5.1	4.0	3.3	2.6	2.0	1.8	1.7	1.7	1.8	2.0	2.3	2.9	3.8	5.6	0.0

** A C O U S T I C R E S U L T S **

REAR ROTOR

PHASE LAG NOT INCLUDED IN RESULTS

HARMONIC # = 1

4.5 FT. ARC PREDICTIONS

OBSERVER ANGLES from INLET re: PROPELLER AXIS

	9.8	19.3	28.5	37.5	46.5	55.5	64.7	74.1	83.8	93.8	104.3	115.1	126.1	137.3	148.4	159.2	169.8
REF dB>	0.0	31.6	58.3	75.1	87.6	98.3	107.3	114.0	118.4	120.5	120.1	116.8	109.8	98.1	79.6	50.1	0.0
PHID				CRP INSTALLATION EFFECT DELTA dB re: Ref dB													
0.0	0.0	5.5	2.5	0.2	-0.9	-0.3	0.2	0.4	0.5	0.5	0.6	0.7	0.9	1.3	2.0	3.7	4.8
15.0	0.0	4.2	0.7	-1.8	-2.5	-1.3	-0.7	-0.4	-0.4	-0.4	-0.3	-0.3	-0.2	0.1	0.7	2.4	3.9
30.0	0.0	2.7	-1.7	-4.7	-4.4	-2.5	-1.6	-1.4	-1.3	-1.3	-1.4	-1.4	-1.5	-1.4	-1.0	0.7	2.8
45.0	0.0	1.0	-5.0	-9.2	-6.4	-3.6	-2.6	-2.4	-2.3	-2.4	-2.5	-2.7	-3.0	-3.3	-3.3	-1.6	1.7
60.0	0.0	-0.5	-9.8	-17.1	-7.5	-4.4	-3.5	-3.3	-3.2	-3.3	-3.6	-4.0	-4.6	-5.5	-6.5	-4.8	0.6
75.0	0.0	-1.1	-11.5	-13.4	-6.8	-4.6	-4.0	-3.8	-3.9	-4.0	-4.4	-4.9	-5.9	-7.7	-11.1	-9.7	0.0
90.0	0.0	-0.4	-6.7	-7.1	-4.9	-4.0	-3.9	-3.9	-4.0	-4.2	-4.6	-5.3	-6.4	-8.6	-14.6	-13.8	0.0
105.0	0.0	1.2	-2.8	-3.4	-2.9	-3.0	-3.3	-3.5	-3.6	-3.8	-4.2	-4.7	-5.7	-7.3	-10.2	-8.8	0.0
120.0	0.0	2.9	-0.1	-0.9	-1.2	-1.9	-2.4	-2.6	-2.8	-3.0	-3.2	-3.6	-4.2	-5.0	-5.8	-4.3	0.7
135.0	0.0	4.4	1.9	0.9	0.2	-0.8	-1.4	-1.7	-1.8	-2.0	-2.1	-2.4	-2.6	-2.9	-2.9	-1.2	1.9
150.0	0.0	5.7	3.4	2.3	1.3	0.2	-0.5	-0.7	-0.9	-1.0	-1.1	-1.1	-1.2	-1.1	-0.7	1.0	3.0
165.0	0.0	6.7	4.5	3.3	2.1	1.0	0.4	0.1	0.0	0.0	-0.1	-0.1	0.0	0.3	0.9	2.6	4.0
180.0	0.0	7.5	5.4	4.1	2.8	1.6	1.1	0.9	0.8	0.8	0.8	0.9	1.1	1.4	2.2	3.9	4.9
195.0	0.0	8.1	6.1	4.7	3.3	2.2	1.6	1.5	1.4	1.4	1.5	1.6	1.9	2.4	3.2	4.9	5.6
210.0	0.0	8.6	6.5	5.1	3.7	2.5	2.1	1.9	1.9	1.9	2.0	2.2	2.6	3.1	3.9	5.6	6.2
225.0	0.0	8.9	6.8	5.4	3.9	2.8	2.4	2.3	2.3	2.3	2.5	2.7	3.1	3.6	4.5	6.2	6.6
240.0	0.0	9.1	7.0	5.5	4.0	3.0	2.6	2.6	2.6	2.6	2.8	3.0	3.4	4.0	4.9	6.6	6.9
255.0	0.0	9.2	7.0	5.5	3.9	3.0	2.7	2.7	2.7	2.8	3.0	3.2	3.6	4.2	5.1	6.8	7.1
270.0	0.0	9.1	6.9	5.3	3.8	2.9	2.7	2.7	2.7	2.8	3.0	3.3	3.7	4.2	5.2	6.9	7.2
285.0	0.0	8.9	6.7	5.0	3.4	2.7	2.6	2.6	2.7	2.8	2.9	3.2	3.6	4.2	5.1	6.8	7.1
300.0	0.0	8.6	6.2	4.5	3.0	2.4	2.4	2.4	2.5	2.6	2.7	3.0	3.4	3.9	4.9	6.5	6.9
315.0	0.0	8.1	5.6	3.8	2.3	2.0	2.0	2.1	2.2	2.2	2.4	2.6	3.0	3.5	4.4	6.1	6.6
330.0	0.0	7.4	4.9	2.9	1.5	1.4	1.6	1.7	1.7	1.8	1.9	2.1	2.4	3.0	3.8	5.5	6.1
345.0	0.0	6.6	3.8	1.8	0.5	0.6	1.0	1.1	1.2	1.2	1.3	1.5	1.8	2.2	3.1	4.7	5.5

** F I N A L C O N S O L I D A T E D A C O U S T I C R E S U L T S **

FRONT ROTOR

PHASE LAG NOT INCLUDED IN RESULTS

HARMONIC # = 1 4.5 FT. SIDELINE

OBSERVER ANGLES from INLET re: PROPELLER AXIS

	10.0	20.0	30.0	40.0	50.0	60.0	70.0	80.0	90.0	100.0	110.0	120.0	130.0	140.0	150.0	160.0	170.0
REF dB>	0.0	9.4	46.7	69.8	84.5	95.4	105.1	112.4	116.5	117.4	115.2	109.3	99.2	83.3	59.1	21.1	0.0
PHID	CRP INSTALLED TONE SPL, dB																
0.0	0.0	15.9	51.0	72.9	87.1	97.3	106.4	113.5	117.5	118.4	116.2	110.5	100.6	85.2	61.9	25.7	0.0
15.0	0.0	14.9	49.9	71.8	86.1	96.5	105.6	112.7	116.7	117.6	115.3	109.5	99.5	83.9	60.6	24.5	0.0
30.0	0.0	13.7	48.4	70.3	84.8	95.4	104.6	111.7	115.7	116.5	114.2	108.3	98.1	82.4	58.9	22.9	0.0
45.0	0.0	12.1	46.4	68.4	83.1	94.0	103.5	110.6	114.5	115.3	112.9	106.8	96.4	80.4	56.7	20.9	0.0
60.0	0.0	10.3	43.5	65.6	80.9	92.4	102.2	109.3	113.3	114.0	111.5	105.2	94.4	77.8	53.5	18.3	0.0
75.0	0.0	8.4	39.2	61.4	78.0	90.6	100.9	108.2	112.2	112.9	110.2	103.6	92.2	74.4	48.1	15.0	0.0
90.0	0.0	7.1	32.4	54.0	74.9	89.0	100.0	107.5	111.6	112.3	109.5	102.7	91.0	71.8	35.1	12.5	0.0
105.0	0.0	7.5	35.0	55.5	74.3	88.4	99.9	107.6	111.8	112.5	109.8	103.2	91.7	73.6	46.3	14.3	0.0
120.0	0.0	9.2	41.0	62.2	77.2	89.5	100.7	108.5	112.6	113.5	110.9	104.6	93.7	77.0	52.5	17.7	0.0
135.0	0.0	11.1	44.6	66.1	80.2	91.3	102.0	109.6	113.8	114.7	112.3	106.3	95.9	79.8	56.1	20.5	0.0
150.0	0.0	12.8	47.1	68.7	82.6	93.1	103.3	110.9	115.0	116.0	113.7	107.8	97.7	82.0	58.5	22.6	0.0
165.0	0.0	14.2	48.9	70.5	84.4	94.6	104.6	112.0	116.2	117.1	114.9	109.2	99.1	83.6	60.3	24.2	0.0
180.0	0.0	15.4	50.2	71.9	85.7	95.9	105.6	113.0	117.1	118.1	115.9	110.3	100.3	84.9	61.7	25.5	0.0
195.0	0.0	16.3	51.3	73.0	86.8	96.9	106.5	113.8	117.9	118.9	116.8	111.1	101.3	85.9	62.7	26.5	0.0
210.0	0.0	17.0	52.1	73.8	87.6	97.7	107.1	114.4	118.5	119.5	117.4	111.9	102.0	86.7	63.5	27.3	0.0
225.0	0.0	17.5	52.6	74.4	88.3	98.3	107.6	114.9	119.0	120.0	117.9	112.4	102.6	87.3	64.1	27.8	0.0
240.0	0.0	17.9	53.0	74.8	88.7	98.7	108.0	115.2	119.3	120.4	118.3	112.8	103.0	87.7	64.5	28.2	0.0
255.0	0.0	18.1	53.3	75.1	89.0	99.0	108.2	115.4	119.5	120.6	118.5	113.0	103.2	88.0	64.8	28.5	0.0
270.0	0.0	18.2	53.4	75.2	89.2	99.2	108.4	115.5	119.6	120.7	118.6	113.1	103.3	88.1	64.9	28.6	0.0
285.0	0.0	18.2	53.4	75.2	89.2	99.2	108.3	115.5	119.6	120.6	118.6	113.0	103.3	88.0	64.8	28.5	0.0
300.0	0.0	18.0	53.3	75.1	89.1	99.1	108.2	115.3	119.4	120.5	118.4	112.8	103.1	87.8	64.6	28.3	0.0
315.0	0.0	17.8	53.0	74.8	88.8	98.9	108.0	115.1	119.1	120.2	118.1	112.5	102.7	87.4	64.2	27.9	0.0
330.0	0.0	17.3	52.5	74.4	88.4	98.5	107.6	114.7	118.7	119.7	117.6	112.0	102.2	86.9	63.6	27.4	0.0
345.0	0.0	16.7	51.8	73.7	87.8	98.0	107.1	114.1	118.2	119.2	117.0	111.4	101.5	86.1	62.9	26.6	0.0

** F I N A L C O N S O L I D A T E D A C O U S T I C R E S U L T S **

REAR ROTOR

PHASE LAG NOT INCLUDED IN RESULTS

HARMONIC # = 1 4.5 FT. SIDELINE

	OBSERVER ANGLES from INLET re: PROPELLER AXIS																
	9.8	19.3	28.5	37.5	46.5	55.5	64.7	74.1	83.8	93.8	104.3	115.1	126.1	137.3	148.4	159.2	169.8
REF dB>	0.0	23.5	53.3	72.0	85.7	97.9	108.2	115.8	120.8	123.2	122.8	119.2	111.5	98.4	77.9	45.1	0.0
PHID	CRP INSTALLED TONE SPL, dB																
0.0	0.0	29.0	55.7	72.2	84.8	97.6	108.4	116.2	121.2	123.7	123.4	119.8	112.3	99.7	79.9	48.8	4.8
15.0	0.0	27.7	54.0	70.1	83.2	96.5	107.5	115.4	120.4	122.8	122.5	118.9	111.3	98.5	78.5	47.5	3.9
30.0	0.0	26.2	51.6	67.3	81.2	95.4	106.5	114.4	119.5	121.9	121.4	117.7	110.0	97.0	76.8	45.8	2.8
45.0	0.0	24.5	48.2	62.7	79.2	94.3	105.5	113.4	118.5	120.8	120.3	116.5	108.5	95.1	74.6	43.5	1.7
60.0	0.0	23.0	43.5	54.9	78.1	93.5	104.7	112.5	117.6	119.9	119.2	115.2	106.9	93.0	71.4	40.3	0.6
75.0	0.0	22.3	41.8	58.6	78.9	93.3	104.2	112.0	116.9	119.2	118.4	114.2	105.5	90.8	66.8	35.4	0.0
90.0	0.0	23.1	46.6	64.9	80.8	93.8	104.3	111.9	116.8	119.0	118.2	113.9	105.1	89.8	63.2	31.3	0.0
105.0	0.0	24.7	50.5	68.6	82.7	94.8	104.9	112.3	117.2	119.4	118.6	114.4	105.8	91.1	67.7	36.3	0.0
120.0	0.0	26.4	53.2	71.1	84.4	96.0	105.8	113.2	118.0	120.2	119.6	115.5	107.3	93.4	72.0	40.9	0.7
135.0	0.0	27.9	55.2	72.9	85.8	97.1	106.8	114.1	119.0	121.2	120.7	116.8	108.8	95.5	75.0	43.9	1.9
150.0	0.0	29.1	56.6	74.2	86.9	98.0	107.7	115.1	119.9	122.2	121.7	118.0	110.3	97.3	77.1	46.1	3.0
165.0	0.0	30.2	57.8	75.3	87.8	98.8	108.5	115.9	120.8	123.1	122.7	119.1	111.5	98.7	78.8	47.7	4.0
180.0	0.0	31.0	58.7	76.1	88.5	99.5	109.2	116.7	121.6	123.9	123.6	120.0	112.5	99.9	80.1	49.0	4.9
195.0	0.0	31.6	59.3	76.7	89.0	100.0	109.8	117.3	122.2	124.6	124.3	120.8	113.4	100.8	81.0	50.0	5.6
210.0	0.0	32.1	59.8	77.1	89.3	100.4	110.2	117.7	122.7	125.1	124.9	121.4	114.0	101.5	81.8	50.7	6.2
225.0	0.0	32.4	60.1	77.3	89.5	100.7	110.6	118.1	123.1	125.5	125.3	121.9	114.5	102.0	82.4	51.3	6.6
240.0	0.0	32.6	60.3	77.5	89.6	100.8	110.8	118.3	123.4	125.8	125.6	122.2	114.9	102.4	82.7	51.7	6.9
255.0	0.0	32.6	60.3	77.4	89.6	100.9	110.9	118.5	123.5	126.0	125.8	122.4	115.1	102.6	83.0	51.9	7.1
270.0	0.0	32.6	60.2	77.3	89.4	100.8	110.9	118.5	123.5	126.0	125.8	122.4	115.1	102.7	83.0	52.0	7.2
285.0	0.0	32.4	59.9	76.9	89.1	100.6	110.8	118.4	123.5	125.9	125.7	122.4	115.0	102.6	83.0	51.9	7.1
300.0	0.0	32.0	59.5	76.5	88.6	100.3	110.6	118.2	123.3	125.8	125.5	122.1	114.8	102.4	82.7	51.6	6.9
315.0	0.0	31.5	58.9	75.8	88.0	99.8	110.2	117.9	123.0	125.4	125.2	121.8	114.4	102.0	82.3	51.2	6.6
330.0	0.0	30.9	58.1	74.9	87.2	99.2	109.7	117.5	122.5	125.0	124.7	121.3	113.9	101.4	81.7	50.6	6.1
345.0	0.0	30.0	57.1	73.7	86.1	98.5	109.1	116.9	122.0	124.4	124.1	120.6	113.2	100.6	80.9	49.9	5.5

CRPCLD Source Code

145 A.

```
C ******************************************************************
C *                                                                *
C *                      CLDCRP   PROGRAM                          *
C *                                                                *
C ******************************************************************
C
C        THIS PROGRAM CALCULATES LIFT AND DRAG COEFFICIENTS
C        AS A FUNCTION OF ANGLE OF ATTACK FOR COUNTER ROTATION
C        PROPELLERS BASED ON SOME OVERALL GEOMETRIC DATA
C        OF THESE PROPELLERS AND CERTAIN MEASURED AERODYNAMIC
C        PERFORMANCE PARAMETERS . THESE RELATIONS ARE OBTAINED
C        ( BASED ON SOME ASSUMPTIONS ) SEPARATELY FOR THE
C        FRONT AND REAR ROTORS .
C
C        IT REQUIRES ACCESS TO THE FOLLOWING 'IMSL' LIBRARY
C        ROUTINES:
C                  CSSCV , CSVAL , NEQNF , PERMU , SVRGP
C
C        INPUT ( in order of appearance in NAMELIST ):
C
C         GEOMETRY  ( One File Per Rotor; Unit 10=R1, Unit 11=R2 ):
C
C                 NZ      = No. of Radial Stations - 10 Max
C                 Z       = NZ values of Radius Ratio (R/Rtip)
C                 CHORD   = NZ values of Blade Chord Normalized by Tip
C                              Diameter - at Z Radial Locations
C                 BETAP   = NZ values of Design, or Reference, Blade Section
C                              Pitch Angle, Degrees - at Z Radial Locations
C                 XPCA    = Rotor Pitch Change Axis Spacing, Normalized by
C                              Rotor 1 Diameter
C                 NB      = No. of Rotor Blades
C                 DT      = Tip Diameter - feet -
C                 HTR     = Hub:Tip Radius Ratio
C
C
C         PERFORMANCE (Unit 12):
C
C                 NP      = No. of Cases for which Aerodynamic
C                              Data are Available : must be .LE. 100
C                                     - integer input -
C                 EM0     = NP values of Forward Flight
C                              Mach No.
C                 PAMB    = NP values of Ambient Static
C                              Pressure : - psia -
C                 TAMB    = NP values of Ambient Static
C                              Temperature : - degrees Rankine -
C                 RPM1    = NP values of Front Rotor RPM
C                 RPM2    = NP values of Rear Rotor RPM
C                 SHP1    = NP values of Front Rotor SHP
C                 SHP2    = NP values of Rear  Rotor SHP
C       ( Either SHP or Torque are Input, depending on the value of ITRQ )
C                 FH1     = NP values of Front Rotor
C                              Thrust - pound force -
C                 FH2     = NP values of Rear Rotor
C                              THRUST - pound force -
C                 BETA341 = NP values of Front Rotor 0.75 R/Rtip
C                              Pitch Angle - degrees -
C                 BETA342 = NP values of Rear Rotor 0.75 R/Rtip
```

```
C                          Pitch Angle - degrees -
C               ITRQ     = 0, Power Input as SHP (Default)
C                        = 1, Power Input as Torque
C               TRQ1     = NP values of Front Rotor Torque
C                                - ft. lbf. -
C               TRQ2     = NP values of Rear Rotor Torque
C                                - ft. lbf. -
C               IAV      = 0, Individual Rotor Diameters and HTRs Used
C                                       (Default)
C                        = 1, Average Values Used
C
C
C          OUTPUT (Unit 6 = Printout):
C             ( of inputs ) :
C                         DT       : Average Diameter - feet -
C                         HTR      : Average Hub:Tip Radius Ratio
C                         CHORD1   : Rotor 1 Chord @ 70% Span -
C                                       Normalized by Rotor 1 Diameter
C                         NB1      : Rotor 1 Blade No.
C                         XPCA
C                         CHORD2   : Rotor 2 Chord @ 70% Span -
C                                       Normalized by Rotor 2 Diameter
C                         NB2      : Rotor 2 Blade No.
C            ( in column form)
C                         EMO , PAMB , TAMB , RPM1 , RPM2 , TRQ1 , TRQ2 ,
C                         SHP1 , SHP2 , FH1 , FH2 , BETA341 , BETA342 .
C
C            ( calculated ) :
C                              1.  TABLE  OF VALUES OF :
C
C                              ALPH1 : Angle of attack  ( front rotor )
C                                          - degrees-
C                               CL1  : Lift coefficient ( front rotor )
C                               CD1  : Drag coefficient ( front rotor )
C                              ALPH2 : Angle of attack  ( rear  rotor )
C                                          - degrees-
C                               CL2  : Lift coefficient ( rear  rotor )
C                               CD2  : Drag coefficient ( rear  rotor )
C
C                              2.  THE DATA OF TABLE ABOVE REORDERED
C                                  SUCH THAT "ALPH1" AND "ALPH2" FORM AN
C                                  INCREASING SEQUENCE . "CL1,CD1,CL2
C                                  AND CD2" ARE CORRESPONDINGLY
C                                  PERMUTED.
C
C                              3.  THE ABOVE DATA " SMOOTHED " USING
C                                  IMSL ROUTINE CSSCV.
C
C
C               (Unit 16 = NAMELIST File for Input to CRPFAN):
C
C                        "$INSTAL"  : Beginning Word
C                        " NPCLCD"  : No. of Data Points
C                        " ALD"     : Smoothed Table of Alpha1
C                        " CL"      : Smoothed Table of CL1
C                        " CD"      : Smoothed Table of CD1
C                        " ALDA"    : Smoothed Table of Alpha2
C                        " CLA"     : Smoothed Table of CL2
```

```
C                           " CDA"      : Smoothed Table of CD2
C
C *******************************************************************
C
C
        COMMON /BLADE1/ Z701,B701,SIG1,TWIST1,DT1,HTR1,B1
C
        COMMON /BLADE2/ Z702,B702,SIG2,TWIST2,DT2,HTR2,B2
C
        COMMON /AERO/ LAM,LAMA,ASEP1,ASEP2,TC1,TC2,QC1,QC2,U1,U2,RRAT,SPR
C
        COMMON /OUTPUT/ AL1,AL2,CLA1,CLA2,CDA1,CDA2
C
C
        DIMENSION Z(10),CHORD(10),BETAP(10),ALE(10),BTE(10),SCD(10)
        DIMENSION THETA(10),THAD(17),TMOC(10),YMC(10),ZMC(10)
        DIMENSION ALPH1(100),CLALP1(100),CDALP1(100)
        DIMENSION ALPH2(100),CLALP2(100),CDALP2(100)
        DIMENSION EM0(100),PAMB(100),TAMB(100),RPM1(100),RPM2(100)
        DIMENSION SHP1(100),SHP2(100),FH1(100),FH2(100)
        DIMENSION BETA341(100),BETA342(100),TRQ1(100),TRQ2(100)
        DIMENSION XI(2),YI(2),IR1(100),IR2(100),IR11(100),IR21(100)
        DIMENSION BRKCL1(100),BRKCD1(100),BRKCL2(100),BRKCD2(100)
        DIMENSION YCL1(100),YCL2(100),YCD1(100),YCD2(100)
        DIMENSION CSCCL1(4,100),CSCCD1(4,100),CSCCL2(4,100),CSCCD2(4,100)
C
        EXTERNAL FSOLV
C
        REAL LAM,LAMA,LAM0,LAM2
C
C       NAMELIST /INPUT/ NZ,Z,CHORD,BETAP,XPCA,NB,DT,HTR
C
      NAMELIST /INPUT/
     & AKINF,ALE,BETA34,BETAP,BETAW,BTE,CHORD,DELBP,DIST,DT,DXTM,
     & HTR,ICOSZ,IDS,IHEL,ILDO,INF,INFS,INST,IPRNTW,IPTO,IRW,ISECT,
     & ISHAPE,ITDO,IWAKE,NALE,NB,NBTE,NCASE,NHM,NP,NSL,NSTEP,NTH,
     & NRSEC,NWHM,NZ,PAMB,RPM,RSEC,SCD,SHP,SIGR,TAMB,THETA,THAD,
     & TMOC,VO,VREF,WTIV,XLM,XPCA,XTM,YMC,Z,ZMC,Z34
C
        NAMELIST /PERFIN/ NP,EM0,PAMB,TAMB,RPM1,RPM2,SHP1,SHP2,
     &             FH1,FH2,BETA341,BETA342,ITRQ,TRQ1,TRQ2,IAV
C
        REPHT = 0.7
C
C
C     READ INPUT DATA FILE (ROTOR 1 GEOMETRY)
C               UNIT 10
C
        READ (10,INPUT,END=9999)
        CLOSE (10)
        PI   = 3.14159
        TPI  = 2.*PI
        DEGR = PI/180.
        RDEG = 1./DEGR
C
C  SET PROGRAM VARIABLES/CONSTANTS - ROTOR 1
C
        DT1     = DT
```

```
      HTR1    = HTR
      NB1     = NB
      B1      = FLOAT(NB)
      NZ1     = NZ
      Z341    = 0.75
      Z701    = HTR1 + REPHT*(1.-HTR1)
      XI(1)   = Z341
      XI(2)   = Z701
      CALL LSPFIT(Z,BETAP,NZ1, XI,YI,2, 0)
      TWIST1 = YI(1)-YI(2)
      CALL LSPFIT(Z,CHORD,NZ1, Z701,CHD701,1, 0)
      SIG1    = B1*CHD701/(PI*Z701)
C
C
C    READ INPUT DATA FILE (ROTOR 2 GEOMETRY)
C             UNIT 11
C
      READ(11,INPUT,END=9999)
      CLOSE (11)
C
C SET PROGRAM VARIABLES/CONSTANTS - ROTOR 2
C
      DT2     = DT
      HTR2    = HTR
      NZ2     = NZ
      NB2     = NB
      B2      = FLOAT(NB)
      Z342    = Z341*DT1/DT2
      Z702    = HTR2 + REPHT*(1.-HTR2)
      XI(1)   = Z342
      XI(2)   = Z702
      CALL LSPFIT(Z,BETAP,NZ2, XI,YI,2, 0)
      TWIST2 = YI(1)-YI(2)
      CALL LSPFIT(Z,CHORD,NZ2, Z702,CHD702,1, 0)
      SIG2    = B2*CHD702/(PI*Z702)
C
C
C     READ PERFORMANCE DATA
C             UNIT 12
C
C
      READ(12,PERFIN,END=9999)
      CLOSE (12)
C
C
C     SET UP QUANTITIES REQU'D BY CODE
C
      RT     = .25*(DT1+DT2)
      DT     = 2.*RT
      HTR    = 0.5*(HTR1*DT1+HTR2*DT2)/DT
      Z70    = HTR + REPHT*(1.-HTR)
      IF (IAV.EQ.1) THEN
          DT1  = DT
          DT2  = DT
          HTR1 = HTR
          HTR2 = HTR
          Z701 = Z70
          Z702 = Z70
```

```
        END IF
        RT1    = .5*DT1
        RT2    = .5*DT2
        RRAT   = RT1/RT2
        ASEP1 = ABS(XPCA)*DT1/RT1
        ASEP2 = ASEP1*RRAT
        AANN   = .25*PI*DT*DT*(1.-HTR*HTR)
        AANN1 = .25*PI*DT1*DT1*(1.-HTR1*HTR1)
        AANN2 = .25*PI*DT2*DT2*(1.-HTR2*HTR2)
        P0     = 14.696*144.
        T0     = 518.67
        RG     = 1716.2
        GAMMA = 1.4
        RHO0   = P0/(RG*T0)
C
C
C *** WRITE OUT INPUT DATA - UNIT 6 ***

        WRITE (6,1010)
        WRITE (6,1020) DT,HTR,CHD701,NB1,XPCA,CHD702,NB2
        WRITE (6,1030)
        WRITE (6,1040)
        WRITE (6,1050)
C
C       INCLUDE MANI CODE
C
        TCON1 = DT1*Z701/(AANN1*RT1**3)
        TCON2 = DT2*Z702/(AANN2*RT2**3)
        QCON1 = TCON1*550./RT1
        QCON2 = TCON2*550./RT2
        DO 100 I=1,NP
            OMEGA1 = TPI*RPM1(I)/60.
            OMEGA2 = TPI*RPM2(I)/60.
            IF(ITRQ.EQ.1) THEN
                QCON1    = TCON1*OMEGA1/RT1
                QCON2    = TCON1*OMEGA2/RT2
                SHP1(I) = TRQ1(I)
                SHP2(I) = TRQ2(I)
            END IF
            IR1(I) = I
            IR2(I) = I
            RHO     = PAMB(I)*144./(TAMB(I)*RG)
            A0      = SQRT(GAMMA*RG*TAMB(I))
            V0      = A0*EM0(I)
            TC1     = FH1(I)*TCON1/(RHO*OMEGA1*OMEGA1)
            QC1     = SHP1(I)*QCON1/(RHO*OMEGA1**3)
            TC2     = FH2(I)*TCON2/(RHO*OMEGA2*OMEGA2)
            QC2     = SHP2(I)*QCON2/(RHO*OMEGA2**3)
            LAM     = V0/(OMEGA1*RT1)
            SPR     = OMEGA1/OMEGA2
            LAMA    = LAM*SPR
            B701    = BETA341(I)-TWIST1
            B702    = BETA342(I)-TWIST2
C
            SHPCN1 = OMEGA1/550.
            SHPCN2 = OMEGA2/550.
C
            IF(ITRQ.EQ.0) THEN
```

```
              WRITE (6,1060) EMO(I),PAMB(I),TAMB(I),RPM1(I),RPM2(I),
     &                 SHP1(I)/SHPCN1,SHP2(I)/SHPCN2,SHP1(I),SHP2(I),
     &                 FH1(I),FH2(I),BETA341(I),BETA342(I)
            ELSE
              WRITE (6,1060) EMO(I),PAMB(I),TAMB(I),RPM1(I),RPM2(I),
     &                 SHP1(I),SHP2(I),SHP1(I)*SHPCN1,SHP2(I)*SHPCN2,
     &                 FH1(I),FH2(I),BETA341(I),BETA342(I)
            END IF

            CALL SOLVER
C
            ALPH1(I)  = AL1
            ALPH2(I)  = AL2
            CLALP1(I) = CLA1
            CDALP1(I) = CDA1
            CLALP2(I) = CLA2
            CDALP2(I) = CDA2
100     CONTINUE
C
C       WRITE (6,1070)
C
C *** END OF PRINCIPAL COMPUTATIONS ***
C
C *** WRITE OUT " RAW DATA " i.e. ALPH1,CLALP1,CDALP1,ALPH2,CLALP2,CDALP2 ***
C
        WRITE (6,1080)
        WRITE (6,1090)
        WRITE (6,1100)
        WRITE (6,1110)
        DO 50 I = 1,NP
          WRITE (6,1120) ALPH1(I),CLALP1(I),CDALP1(I),
     &                   ALPH2(I),CLALP2(I),CDALP2(I)
   50   CONTINUE
C
C *** REORDER IN INCREASING "ALPH1 , ALPH2" AND PERMUTE " CL1,CD1,CL2,CD2"
C *** ACCORDINGLY ***
C
C
        CALL SVRGP (NP,ALPH1, ALPH1,IR1)
        CALL SVRGP (NP,ALPH2, ALPH2,IR2)
        CALL PERMU (NP,CLALP1,IR1,1,CLALP1)
        CALL PERMU (NP,CDALP1,IR1,1,CDALP1)
        CALL PERMU (NP,CLALP2,IR2,1,CLALP2)
        CALL PERMU (NP,CDALP2,IR2,1,CDALP2)
C
C *** SMOOTH DATA AND EVALUATE AT " ALPH1,ALPH2 "***
C
C
        IEQUAL = 2
        CALL CSSCV(NP,ALPH1,CLALP1,IEQUAL, BRKCL1,CSCCL1)
        CALL CSSCV(NP,ALPH1,CDALP1,IEQUAL, BRKCD1,CSCCD1)
        CALL CSSCV(NP,ALPH2,CLALP2,IEQUAL, BRKCL2,CSCCL2)
        CALL CSSCV(NP,ALPH2,CDALP2,IEQUAL, BRKCD2,CSCCD2)
C
        DO 70 IP = 1,NP
            YCL1(IP) = CSVAL(ALPH1(IP),NP-1,BRKCL1,CSCCL1)
            YCD1(IP) = CSVAL(ALPH1(IP),NP-1,BRKCD1,CSCCD1)
            YCL2(IP) = CSVAL(ALPH2(IP),NP-1,BRKCL2,CSCCL2)
```

```
          YCD2(IP) = CSVAL(ALPH2(IP),NP-1,BRKCD2,CSCCD2)
   70   CONTINUE
C
C *** WRITE OUT " REORDERED " AND " SMOOTHED " DATA
        WRITE (6,1080)
        WRITE (6,1090)
        WRITE (6,1130)
        WRITE (6,1140)
        DO 80 I = 1,NP
          WRITE (6,1120) ALPH1(I),CLALP1(I),CDALP1(I),ALPH2(I),CLALP2(I),
     &                   CDALP2(I)
   80   CONTINUE
        WRITE (6,1080)
        WRITE (6,1090)
        WRITE (6,1150)
        WRITE (6,1160)
        DO 90 I = 1,NP
          WRITE (6,1120) ALPH1(I),YCL1(I),YCD1(I),ALPH2(I),YCL2(I),YCD2(I)
   90   CONTINUE
C
C       WRITE NAMELIST FILE FOR INPUT TO CRPIE
C
        WRITE (16,2000)
        WRITE (16,2010) NP
        WRITE (16,2020)
        WRITE (16,2030) (ALPH1(I),I=1,NP)
        WRITE (16,2040)
        WRITE (16,2050) (YCL1(I),I=1,NP)
        WRITE (16,2060)
        WRITE (16,2050) (YCD1(I),I=1,NP)
        WRITE (16,2070)
        WRITE (16,2030) (ALPH2(I),I=1,NP)
        WRITE (16,2080)
        WRITE (16,2050) (YCL2(I),I=1,NP)
        WRITE (16,2090)
        WRITE (16,2050) (YCD2(I),I=1,NP)
C
C *** FORMAT STATEMENTS ***-----------------------------------------
C
 1000 FORMAT(2X,'INPUT ERROR - NP EXCEEDS 100, GOING TO NEXT CASE')
 1010 FORMAT(20X,'***** I N P U T   P A R A M E T E R S *****',//)
 1020 FORMAT(20X,'TIP DIAMETER  ( AVG. OF BOTH ROTORS ) - ft. =',F8.4,/,
     &       20X,'HUB:TIP RATIO ( AVG. OF BOTH ROTORS )       =',F8.4,/,
     &       20X,' 70 % SPAN CHORD:DIAMETER ( FRONT ROTOR )   =',F8.4,/,
     &       20X,'NUMBER OF BLADES ( FRONT ROTOR )            =',I4,/,
     &       20X,' (PCA SPACING):(ROTOR 1 DIAMETER) RATIO     =',F8.4,/,
     &       20X,' 70 % SPAN CHORD:DIAMETER ( REAR  ROTOR )   =',F8.4,/,
     &       20X,'NUMBER OF BLADES ( REAR  ROTOR )            =',I4,//)
 1030 FORMAT(20X,'NOTATION AND UNITS FOR INPUT DATA TABLE',//,
     &       20X,'EM0  : FORWARD FLIGHT MACH NUMBER',/,
     &       20X,'PAMB : AMBIENT PRESSURE - psia',/,
     &       20X,'TAMB : AMBIENT TEMPERATURE - degrees Rankine',/,
     &       20X,'RPM1 : RPM OF FRONT ROTOR',/,
     &       20X,'RPM2 : RPM OF REAR  ROTOR',/,
     &       20X,'TRQ1 : TORQUE OF FRONT ROTOR - ft. lbf',/,
     &       20X,'TRQ2 : TORQUE OF REAR  ROTOR - ft. lbf',/,
     &       20X,'SHP1 : POWER ABSORBED BY FRONT ROTOR - SHP',/,
     &       20X,'SHP1 : POWER ABSORBED BY REAR  ROTOR - SHP',/,
```

```
     &        20X,'THRF : THRUST OF FRONT ROTOR - lbf',/,
     &        20X,'THRR : THRUST OF REAR  ROTOR - lbf',/,
     &        20X,'B341 : 0.75 RADIUS RATIO PITCH ( deg.) - FRONT',/,
     &        20X,'B342 : 0.75 RADIUS RATIO PITCH ( deg.) - REAR ')
 1040 FORMAT('1',//,20X,'   I N P U T   D A T A   T A B L E ',//)
 1050 FORMAT(2X,' EMO ',1X,' PAMB ',1X,' TAMB ',1X,'RPM1 ',1X,
     &'RPM2 ',1X,' TRQ1 ',1X,' TRQ2 ',1X,' SHP1 ',1X,' SHP2 ',1X,
     &' THRF ',1X,' THRR ',1X,'B341 ',1X,'B342 ',/)
 1060 FORMAT(2X,F5.3,1X,F6.3,1X,F6.2,1X,F5.0,1X,F5.0,F6.1,1X,F6.1,
     &1X,F6.1,1X,F6.1,1X,F6.1,1X,F6.1,1X,F5.2,1X,F5.2)
 1070 FORMAT(//,20X,'*** END WRITE OF INPUT PARAMETERS***')
 1080 FORMAT('1',///,20X,' C A L C U L A T E D   Q U A N T I T I E S')
 1090 FORMAT(//,25X,'Abbreviations : AA : Angle of attack',/,
     &        25X,'                F : Front',/,
     &        25X,'                R : Rear ',/)
 1100 FORMAT(/,2X,5X,'AA:F',6X,'CL:F',6X,'CD:F',6X,'AA:R',6X,
     &'CL:R',6X,'CD:R',' (RAW DATA)')
 1110 FORMAT(5X,'( deg. )',22X,'( deg. )',/)
 1120 FORMAT(2X,6F10.5)
 1130 FORMAT(/,2X,5X,'AA:F',6X,'CL:F',6X,'CD:F',6X,'AA:R',6X,
     &'CL:R',6X,'CD:R',' (REORDERED ')
 1140 FORMAT(5X,'( deg. )',22X,'( deg. )',23X,'RAW DATA)',/)
 1150 FORMAT(/,2X,5X,'AA:F',6X,'CL:F',6X,'CD:F',6X,'AA:R',6X,
     &'CL:R',6X,'CD:R',' (REORDERED &')
 1160 FORMAT(5X,'( deg. )',22X,'( deg. )',23X,'SMOOTHED)',/)
C
C *** FORMATS FOR NAMELIST OUTPUT FILE ***
C
 2000 FORMAT(1X,'$INSTAL,')
 2010 FORMAT(1X,'NPCLCD=',I3,',')
 2020 FORMAT(1X,'ALD=')
 2030 FORMAT(3X,8(F7.3,','))
 2040 FORMAT(1X,'CL=')
 2050 FORMAT(3X,8(F8.4,','))
 2060 FORMAT(1X,'CD=')
 2070 FORMAT(1X,'ALDA=')
 2080 FORMAT(1X,'CLA=')
 2090 FORMAT(1X,'CDA=')
C
C-----------------------------------------------------------------------
C
 9999   CONTINUE
C
        STOP
        END
        SUBROUTINE SOLVER


C       FUNCTIONAL DESCRIPTION:
C
C            CALLS IMSL ROUTINES TO SOLVE C/R PROPELLER PERFORMANCE
C            CL & CD VS ALPHA FORMULATION
C
C


        DIMENSION X(4),XG(4)

        REAL LAM,LAMA
```

```
C
        COMMON /BLADE1/ Z701,B701,SIG1,TWIST1,DT1,HTR1,B1
C
        COMMON /BLADE2/ Z702,B702,SIG2,TWIST2,DT2,HTR2,B2
C
        COMMON /AERO/ LAM,LAMA,ASEP1,ASEP2,TC1,TC2,QC1,QC2,U1,U2,RRAT,SPR
C
        COMMON /OUTPUT/ AL1,AL2,CLA1,CLA2,CDA1,CDA2
C
C
        EXTERNAL FSOLV

        PI   = 3.14159
        TPI  = 2.*PI
        DEGR = PI/180.
        RDEG = 1./DEGR
C
C       START OF COMPUTATIONS
C
        AS      = ASEP1/Z701
        U1      = 1.+AS/SQRT(AS**2+1.)
        AS      = ASEP2/Z702
        U2      = 1.-AS/SQRT(AS**2+1.)
        N       = 4
        RK      = TC2/(TC1*SPR**2*RRAT**4)
        T1      = 1./SQRT(1.+RK)
        T2      = SQRT(.25*TC1/(Z701*LAM**2)+.25/(1.+RK))
        A       = T1*(T2-.5*T1)
        AA      = RK*A
        X(3)    = A
        X(4)    = AA
        T1      = .25/(1.+A+AA)
        X3      = QC1*T1/(LAM*Z701**3)
        X4      = QC2*T1/(LAMA*Z702**3)
        TX1     = LAM*(1.+A+AA)/(Z701*(1.-X3))
        TX2     = LAMA*(1.+A+AA)/(Z702*(1.+2.*X3*SPR-X4))
        X(1)    = B701-ATAN(TX1)/DEGR
        X(2)    = B702-ATAN(TX2)/DEGR
        ITMAX   = 200
        ERRREL  = .0001
        DO 10 IG=1,4
            XG(IG)=X(IG)
10      CONTINUE
C
C       CALL IMSL ROUTINE
C
        CALL NEQNF(FSOLV,ERRREL,N,ITMAX,XG, X,FNORM)
C
        PHID1    = B701-X(1)
        PHID2    = B702-X(2)
        PHIR1    = PHID1*DEGR
        PHIR2    = PHID2*DEGR
        SPH1     = SIN(PHIR1)
        CPH1     = COS(PHIR1)
        TPH1     = SPH1/CPH1
        SECPH21  = 1./(CPH1**2)
        SPH2     = SIN(PHIR2)
        CPH2     = COS(PHIR2)
```

```
      TPH2     = SPH2/CPH2
      SECPH22 = 1./(CPH2**2)
      AA1      = X(3)
      AA2      = X(4)
      FLC      = .5*B1*(1.-Z701)/(Z701*SPH1)
      PTLF     = (2./PI)*ACOS(EXP(-FLC))
      FLC      = .5*B2*(1.-Z702)/(Z702*SPH2)
      PTLFA    = (2./PI)*ACOS(EXP(-FLC))
C
      APF      = 1.-(LAM*(1.+AA1+PTLFA*AA2*U2)/(TPH1*Z701))
      APR      = 1.-(LAMA*(1.+AA2+PTLF*AA1*U1)/(TPH2*Z702)-2.*PTLF*APF*SPR)
C
      Q3M      = SIG1/(4.*SPH1*CPH1*PTLF)
      CX1      = (APF/(1.-APF))/Q3M
      Q3       = Q3M*CX1
      Q4M      = SIG2/(4.*SPH2*CPH2*PTLFA)
      CX2      = (APR/(1.+2.*PTLF*APF*SPR-APR))/Q4M
      Q4       = Q4M*CX2
      Q1M      = (Q3/CX1)*Z701/LAM
      CY1      = AA1/(Q1M*(1.-APF))
      Q1       = Q1M*CY1
      Q2M      = (Q4/CX2)*Z702/LAMA
      CY2      = AA2/((1.+2.*PTLF*APF*SPR-APR)*Q2M)
      AL1      = X(1)
      AL2      = X(2)
      CLA1     = CX1*SPH1+CY1*CPH1
      CDA1     = -CY1*SPH1+CX1*CPH1
      CLA2     = CX2*SPH2+CY2*CPH2
      CDA2     = -CY2*SPH2+CX2*CPH2
      RETURN
      END
C*********************************************************************
      SUBROUTINE FSOLV(X,F,N)


C     FUNCTIONAL DESCRIPTION:
C
C             ROUTINE CALLED BY IMSL NEQNF FROM SUBROUTINE SOLVER
C
C

      DIMENSION X(4),F(4)

      REAL LAM,LAMA
C
      COMMON /BLADE1/ Z701,B701,SIG1,TWIST1,DT1,HTR1,B1
C
      COMMON /BLADE2/ Z702,B702,SIG2,TWIST2,DT2,HTR2,B2
C
      COMMON /AERO/ LAM,LAMA,ASEP1,ASEP2,TC1,TC2,QC1,QC2,U1,U2,RRAT,SPR
C
      COMMON /OUTPUT/ AL1,AL2,CLA1,CLA2,CDA1,CDA2
C

      PI   = 3.14159
      TPI  = 2.*PI
      DEGR = PI/180.
      RDEG = 1./DEGR
```

```
C
C       START OF COMPUTATIONS
C
        PHID1   = B701-X(1)
        PHID2   = B702-X(2)
        PHIR1   = PHID1*DEGR
        PHIR2   = PHID2*DEGR
        SPH1    = SIN(PHIR1)
        CPH1    = COS(PHIR1)
        TPH1    = SPH1/CPH1
        SECPH21 = 1./(CPH1**2)
        SPH2    = SIN(PHIR2)
        CPH2    = COS(PHIR2)
        TPH2    = SPH2/CPH2
        SECPH22 = 1./(CPH2**2)
        AA1     = X(3)
        AA2     = X(4)
        FLC     = .5*B1*(1.-Z701)/(Z701*SPH1)
        PTLF    = (2./PI)*ACOS(EXP(-FLC))
        FLC     = .5*B2*(1.-Z702)/(Z702*SPH2)
        PTLFA   = (2./PI)*ACOS(EXP(-FLC))
C
        APF     = 1.-(LAM*(1.+AA1+PTLFA*AA2*U2)/(TPH1*Z701))
        APR     = 1.-(LAMA*(1.+AA2+PTLF*AA1*U1)/(TPH2*Z702)-2.*PTLF*APF*SPR)
C
        Q3M     = SIG1/(4.*SPH1*CPH1*PTLF)
        CX1     = (APF/(1.-APF))/Q3M
        Q3      = Q3M*CX1
        Q4M     = SIG2/(4.*SPH2*CPH2*PTLFA)
        CX2     = (APR/(1.+2.*PTLF*APF*SPR-APR))/Q4M
        Q4      = Q4M*CX2
        Q1M     = (Q3/CX1)*Z701/LAM
        CY1     = AA1/(Q1M*(1.-APF))
        Q1      = Q1M*CY1
        Q2M     = (Q4/CX2)*Z702/LAMA
        CY2     = AA2/((1.+2.*PTLF*APF*SPR-APR)*Q2M)
        Q2      = Q2M*CY2
        AP1     = APF
        AP2     = APR
        TF1     = SIG1*Z701**3*(1.-AP1)**2*CY1*SECPH21
        TF2     = SIG2*Z702**3*(1.+2.*AP1*SPR*PTLF-AP2)**2*CY2*SECPH22
        QF1     = TF1*Z701*CX1/CY1
        QF2     = TF2*Z702*CX2/CY2
        F(1)    = TF1-TC1
        F(2)    = TF2-TC2
        F(3)    = QF1-QC1
        F(4)    = QF2-QC2
        RETURN
        END
C
C ********************************
C *      SUBROUTINE LSPFIT       *
C ************************************************************************
C
C     THIS ROUTINE IS A LEAST-SQUARES PARABOLIC CURVE FIT PROGRAM THAT
C     WILL INTEGRATE OR INTERPOLATE USING A PARABOLA WHICH PASSES THROUGH
C     POINTS I AND I+1 AND MISSES POINTS I-1 AND I+2 (IF THEY BOTH
C     EXIST) SUCH THAT THE SQUARE OF THE DEVIATION IS A MINIMUM.
```

```
C      NOTE THAT  I  IS GENERALLY SELECTED SUCH THAT
C                 X(I).LE.XC.LT.X(I+1)
C      THE EQUATION FOR THE PARABOLA IS:
C                 Y-Y(I)  =  B*(X-X(I)) + C*(X-X(I))**2
C      OUTSIDE OF THE X,Y-DATA RANGE, LINEAR EXTRAPOLATION OF THE
C      PARABOLA END POINT SLOPE IS EMPLOYED IF NXTRP = 0......29 AUG 85
C
C      INPUT:
C        X, Y      = PTS. ON CURVE
C        NPTS      = NO. OF X    (NPTS > 0)
C        XC        = LIST OF X AT WHICH CALC TO BE DONE
C        YC(1)     = INTEGRATION CONSTANT IF  ND = -1
C        NXC       =  NO. OF XC
C        ND        = 0 TO GET COORD,   = 1 TO GET 1ST DERIVATIVE,
C                  = -1 FOR INTEGRATION   (NPTS > 2)
C        LEND      =  LINEAR FIT IN END INTERVAL, T OR F
C        NXTRP = T OR F TO USE Y-TABLE END VALUES OR EXTRAPOLATE
C
C      OUTPUT:
C        YC        = COORDINATE OR DERIVATIVE AT XC    OR
C        YC(IC)    = INTEGRAL(Y*DX) FROM XC(1) TO XC(IC) WHERE IC = 2,NXC
C
C      NOTES
C           'X' MAY BE IN EITHER ASCENDING OR DESCENDING ORDER.
C           FOR INTEGRATION 'XC' MUST BE IN THE SAME ORDER AS 'X';
C           FOR INTERPOLATION, NO SPECIAL ORDER IS REQIRED.
C
C ***********************************************************************
C
      SUBROUTINE LSPFIT(X,Y,NPTS, XC,YC,NXC, ND)
C
      LOGICAL              LEND
      LOGICAL          NXTRP
      LOGICAL          WITHIN
C
      DIMENSION X(1),Y(1), XC(1),YC(1)
C
      COMMON /CLSPF / INVL,LEND
      COMMON /CLSPF2/ NXTRP
C
      DATA INVL,LEND,NXTRP/0,.FALSE.,.FALSE./
C
      N      = NPTS-1
      IF(ND.EQ.(-1)) INVL = 0
      ISAVE = -1
      SGN    = SIGN(1.,X(N+1)-X(1))
C
C *** BEGIN LOOP FOR EVALUATING YC(IC) AT XC(IC),  IC = 1,NXC
C
      IC     = 1
C
C *** LOCATE APPROPRIATE INTERVAL, INVL
C ***
C *** I      = INTERVAL INDEX WITHIN DATA LIMITS  (1....N)
C *** INVL   = EXPANDED INTERVAL INDEX  (0....N+1),
C ***   WHERE BY IMPLICATION THE ZEROTH INTERVAL IS
C ***   FROM  -INFINITY TO X(1)  AND  THE N+1 INTERVAL IS
C ***   FROM  X(NPTS) TO +INFINITY.  HENCE, AN INTERVAL, INVL,
```

```
C ***   CAN ALWAYS BE FOUND WHICH BOUNDS  XC(IC).
C
 100  INVL   = MAX(0,MIN(INVL,N+1))
      WITHIN = .FALSE.
      NCOUNT = N+2
 110  IF(NCOUNT) 200,120,120
 120  NCOUNT = NCOUNT-1
C
C *** CHECK FOR JUST ONE INPUT POINT
C *** IF(N.EQ.0)     GO TO 220
C *** CHECK IF XC IS TO THE RIGHT OF THE INTERVAL'S LEFT BOUND
C
      IF(INVL.EQ.0)  GO TO 160
      XI = X(INVL)
      XD = XC(IC)-XI
 130  IF(SGN*XD) 140,150,160
C
C *** F.LT.0. (F IS THE FRACTIONAL POSITION IN THE INTERVAL)
C
 140  IF(ND.EQ.(-1)) GO TO 200
      INVL = INVL-1
      GO TO 110
C
C *** F.EQ.0
C *** IF(X(INVL+1).NE.XI) GO TO 210 GGA 10/09/85 DONT CHECK IF BEYOND END
C
 150  IF(INVL.GE.NPTS.OR.X(INVL+1).NE.XI) GO TO 210
      IF(INVL.GT.N) GO TO 140
      GO TO 190
C
C *** F.GT.0.
C *** CHECK IF XC IS TO THE LEFT OF THE INTERVAL'S RIGHT BOUND
C
 160  IF(INVL.EQ.NPTS) GO TO 210
      IF(SGN*(XC(IC)-X(INVL+1))) 210,170,180
C
C *** F.EQ.1.0,  CHK FOR INTEGRATION AND DOUBLE PT BEFOR INCREMENTING INVL
C
 170  IF( (ND.EQ.(-1)) .OR.
     & (INVL.NE.N .AND. X(INVL+1).EQ.X(INVL+2)) ) GO TO 210
C
C *** F.GT.1.0
C
 180  IF(ND.EQ.(-1)) GO TO 220
 190  INVL = INVL+1
      GO TO 110
C
C *** UNSUCCESSFUL SEARCH FOR APPROPRIATE INTERVAL
C
 200  CONTINUE
C
 210  WITHIN = .TRUE.
C
C *** LEAST SQUARE PARABOLA FIT  FOR INTERVAL INVL
C
 220  IF(INVL-ISAVE) 230,300,230
 230  ISAVE = INVL
      I     = MAX(1,MIN(INVL,N))
```

```
      XI     = X(I)
      XD     = XC(IC)-XI
      YI     = Y(I)
      B      = 0.
      C      = 0.
      TOP    = 0.
      BOT    = 0.
      IF(N.EQ.0) GO TO 300
      X3     = X(I+1)-XI
      Y3     = Y(I+1)-YI
      IF(LEND .AND. (I.EQ.1 .OR. I.EQ.N)) GO TO 270
      IF(I.EQ.1) GO TO 240
      X1     = X(I-1)-XI
      X13    = X(I-1)-X(I+1)
      TOP    = X1*(Y3*X1-(Y(I-1)-YI)*X3)*X13
      BOT    = (X1*X13)*(X1*X13)*X3
 240   IF(I.GE.N) GO TO 260
      IF(XD.EQ.0. .AND. BOT.NE.0.) GO TO 250
      X4     = X(I+2)-XI
      X43    = X(I+2)-X(I+1)
      Y4     = Y(I+2)-YI
      TOP    = TOP + X4*(Y3*X4-Y4*X3)*X43
      BOT    = BOT + (X4*X43)*(X4*X43)*X3
      GO TO 260
 250  ISAVE = 0
C
C *** (X1**2 + X43**2)  MUST BE GREATER THAN  (X3/1000)**2
C
 260  IF(BOT.NE.0. .AND. ABS(BOT).GE.ABS((X3*X3)*(X3*X3)*X3*1.E-6))
     & C     = -TOP/BOT
 270   IF(X3.NE.0.) B = (Y(I+1)-YI)/X3 - C*X3
C
C *** IF XC IS OUTSIDE OF X-RANGE, REDEFINE CURVE PARAMETERS
C *** TO EXTRAPOLATE AT END POINT SLOPE.
C
      IF(I.EQ.INVL) GO TO 300
      IF(INVL.EQ.0) GO TO 290
C
C *** RIGHT END EXTRAPOLATION, ADJUST B, YI AND XD
C
      B  = B+2*C*X3
      YI = Y(I+1)
      XI = X(I+1)
 290  C      =  0.
      IF(NXTRP) B = 0.
C
C *** INTEGRATION, INTERPOLATION OR DIFFERENTIATION, STORE RESULT IN YC(IC)
C
 300  XD = XC(IC)-XI
      IF(ND) 310,340,350
C
C *** ND = -1, INTEGRATE
C
 310  IF(.NOT.WITHIN) XD = X(INVL+1)-XI
      S1 = (YI + (B/2. + C/3.*XD)*XD)*XD
      IF(WITHIN) GO TO 320
C
C *** 'INVL' IS BEING INCREMENTED TO FIND APPROPRIATE INTERVAL.  HENCE,
```

```
C *** CUMULATE THE INTEGRAL OF THE THIS INTERVAL.
C
      SA = SA + S1
      GO TO 190
C
C *** APPROPRIATE INTERVAL FOUND.   X(INVL)-XC(IC)-X(INVL+1)
C
 320  IF(IC.NE.1) GO TO 330
C
C *** LEFT LIMIT OF INTEGRATION
C
      SA = YC(IC)-S1
      GO TO 360
C
C *** RIGHT LIMIT OF INTEGRATION
C
 330  YC(IC) = SA+S1
      GO TO 360
C
C *** ND = 0,  INTERPOLATE FOR COORDINATES
C
 340  YC(IC) =  YI + (B + C*XD)*XD
      GO TO 360
C
C *** ND = 1,  FIRST DERIVATIVE
C
 350  YC(IC) =  B + 2.*C*XD
      GO TO 360
C
C *** END OF LOOP, INDEX IC
C
 360  IC = IC+1
      IF(NXC-IC) 380,370,370
 370  IF(ND.NE.(-1).AND.XC(IC).EQ.XC(IC-1)) INVL = INVL+1
      GO TO 100
C
 380  CONTINUE
C
      RETURN
      END
```

Sample CRPCLD Input File:

Rotor 1 Geometry

```
$INPUT
DT=2.0538 HTR=0.4247  NB=11
XPCA=.2394
NZ=10
Z=
1. .9793 .9584 .9153 .8234 .7199 .5957 .5198 .4757 .4247
CHORD=
.0330 .0555 .0704 .0877 .1057 .1205 .1281 .1277 .1266 .1248
BETAP=
43.92 46.54 47.78 49.54 53.41 58.20 64.80 69.10 70.52 70.42
$
```

Sample CRPCLD Input File:

Rotor 2 Geometry

```
$INPUT
DT=1.9952 HTR=0.4148  NB=9
NZ=10
Z=
1. .9779 .9561 .9115 .8173 .7121 .5866 .5094 .4652 .4148
CHORD=
.0348 .0585 .0736 .0913 .1096 .1245 .1311 .1312 .1307 .1296
BETAP=
45.41 47.38 47.93 49.39 52.10 55.57 60.31 63.60 64.52 64.26
$
```

Sample CRPCLD Input File:

Performance Input

```
$PERFIN
NP=11 IAV=1
EM0=
  .1995 .1998 .2004 .1997 .1995 .2004 .2006 .2000 .1998
  .2007 .2008
PAMB=
  13.7753 13.7722 13.7723 13.7682 13.7647 13.7603 13.7553
  13.7553 13.7528 13.7473 13.7473
TAMB=
  532.35 532.63 532.58 531.70 533.77 532.04 532.28 532.70 531.37
  533.77 535.03
RPM1=
  5515 5946 5943 6365 6795 7201 7214 7638 7633 8054 8063
RPM2=
  5567 5998 5995 6417 6849 7261 7261 7690 7695 8109 8111
ITRQ=0
SHP1=
  60.99 84.86 84.82 112.45 146.29 185.77 186.23 229.09
  229.82 277.59 278.34
SHP2=
  62.57 86.89 87.36 117.08 151.47 192.01 192.41 238.54
  239.35 287.97 288.41
FH1=
  112.97 146.69 147.78 185.66 228.73 273.88 274.55 319.39
  321.07 368.64 369.73
FH2=
   90.94 122.65 122.93 154.55 187.66 228.30 228.56 272.12
  270.49 312.73 312.03
BETA341=
  11*36.4
BETA342=
  11*36.5
$
```

Sample CRPCLD Output File:

Printout

```
***** I N P U T   P A R A M E T E R S *****

TIP DIAMETER  ( AVG. OF BOTH ROTORS ) - ft. =  2.0245
HUB:TIP RATIO ( AVG. OF BOTH ROTORS )       =  0.4198
 70 % SPAN CHORD:DIAMETER ( FRONT ROTOR )   =  0.1050
NUMBER OF BLADES ( FRONT ROTOR )            =  11
 (PCA SPACING):(ROTOR 1 DIAMETER) RATIO     =  0.2394
 70 % SPAN CHORD:DIAMETER ( REAR  ROTOR )   =  0.1084
NUMBER OF BLADES ( REAR  ROTOR )            =   9


NOTATION AND UNITS FOR INPUT DATA TABLE

EM0  : FORWARD FLIGHT MACH NUMBER
PAMB : AMBIENT PRESSURE - psia
TAMB : AMBIENT TEMPERATURE - degrees Rankine
RPM1 : RPM OF FRONT ROTOR
RPM2 : RPM OF REAR  ROTOR
TRQ1 : TORQUE OF FRONT ROTOR - ft. lbf
TRQ2 : TORQUE OF REAR  ROTOR - ft. lbf
SHP1 : POWER ABSORBED BY FRONT ROTOR - SHP
SHP1 : POWER ABSORBED BY REAR  ROTOR - SHP
THRF : THRUST OF FRONT ROTOR - lbf
THRR : THRUST OF REAR  ROTOR - lbf
B341 : 0.75 RADIUS RATIO PITCH ( deg.) - FRONT
B342 : 0.75 RADIUS RATIO PITCH ( deg.) - REAR
```

I N P U T   D A T A   T A B L E

EMO	PAMB	TAMB	RPM1	RPM2	TRQ1	TRQ2	SHP1	SHP2	THRF	THRR	B341	B342
0.199	13.775	532.35	5515.	5567.	58.1	59.0	61.0	62.6	113.0	90.9	36.40	36.50
0.200	13.772	532.63	5946.	5998.	75.0	76.1	84.9	86.9	146.7	122.7	36.40	36.50
0.200	13.772	532.58	5943.	5995.	75.0	76.5	84.8	87.4	147.8	122.9	36.40	36.50
0.200	13.768	531.70	6365.	6417.	92.8	95.8	112.4	117.1	185.7	154.6	36.40	36.50
0.199	13.765	533.77	6795.	6849.	113.1	116.2	146.3	151.5	228.7	187.7	36.40	36.50
0.200	13.760	532.04	7201.	7261.	135.5	138.9	185.8	192.0	273.9	228.3	36.40	36.50
0.201	13.755	532.28	7214.	7261.	135.6	139.2	186.2	192.4	274.5	228.6	36.40	36.50
0.200	13.755	532.70	7638.	7690.	157.5	162.9	229.1	238.5	319.4	272.1	36.40	36.50
0.200	13.753	531.37	7633.	7695.	158.1	163.4	229.8	239.4	321.1	270.5	36.40	36.50
0.201	13.747	533.77	8054.	8109.	181.0	186.5	277.6	288.0	368.6	312.7	36.40	36.50
0.201	13.747	535.03	8063.	8111.	181.3	186.8	278.3	288.4	369.7	312.0	36.40	36.50

C A L C U L A T E D   Q U A N T I T I E S

Abbreviations : AA : Angle of attack
F : Front
R : Rear

AA:F (deg.)	CL:F	CD:F	AA:R (deg.)	CL:R	CD:R	(RAW DATA)
1.85037	0.36043	0.00347	3.70063	0.29824	0.03492	
2.41850	0.41292	0.00694	4.11853	0.33964	0.03477	
2.33315	0.41473	0.00500	4.02207	0.34078	0.03491	
2.90747	0.46288	0.00719	4.44438	0.37158	0.04009	
3.29176	0.51119	0.00891	4.75683	0.39456	0.04442	
3.54384	0.55249	0.01224	4.95176	0.42015	0.04524	
3.55883	0.55206	0.01195	4.91594	0.42088	0.04524	
3.96224	0.58277	0.01640	5.20340	0.44329	0.04630	
3.96597	0.58540	0.01615	5.25399	0.44037	0.04823	
4.20923	0.61371	0.01873	5.40352	0.45628	0.04843	
4.18850	0.61527	0.01819	5.37027	0.45618	0.04892	

C A L C U L A T E D   Q U A N T I T I E S

Abbreviations : AA : Angle of attack
F : Front
R : Rear

AA:F (deg.)	CL:F	CD:F	AA:R (deg.)	CL:R	CD:R (REORDERED RAW DATA)
1.85037	0.36043	0.00347	3.70063	0.29824	0.03492
2.33315	0.41473	0.00500	4.02207	0.34078	0.03491
2.41850	0.41292	0.00694	4.11853	0.33964	0.03477
2.90747	0.46288	0.00719	4.44438	0.37158	0.04009
3.29176	0.51119	0.00891	4.75683	0.39456	0.04442
3.54384	0.55249	0.01224	4.91594	0.42088	0.04524
3.55883	0.55206	0.01195	4.95176	0.42015	0.04524
3.96224	0.58277	0.01640	5.20340	0.44329	0.04630
3.96597	0.58540	0.01615	5.25399	0.44037	0.04823
4.18850	0.61527	0.01819	5.37027	0.45618	0.04892
4.20923	0.61371	0.01873	5.40352	0.45628	0.04843

C A L C U L A T E D   Q U A N T I T I E S

Abbreviations : AA : Angle of attack
F : Front
R : Rear

AA:F (deg.)	CL:F	CD:F	AA:R (deg.)	CL:R	CD:R	(REORDERED & SMOOTHED)
1.85037	0.35706	0.00353	3.70063	0.30359	0.03307	
2.33315	0.40986	0.00559	4.02207	0.33290	0.03610	
2.41850	0.41920	0.00591	4.11853	0.34169	0.03700	
2.90747	0.47267	0.00740	4.44438	0.37140	0.04007	
3.29176	0.51470	0.00957	4.75683	0.39988	0.04302	
3.54384	0.54226	0.01185	4.91594	0.41439	0.04452	
3.55883	0.54390	0.01200	4.95176	0.41765	0.04486	
3.96224	0.58801	0.01613	5.20340	0.44059	0.04723	
3.96597	0.58842	0.01617	5.25399	0.44521	0.04770	
4.18850	0.61275	0.01841	5.37027	0.45581	0.04880	
4.20923	0.61502	0.01862	5.40352	0.45884	0.04911	

Sample CRPCLD Output File:

Namelist

```
$INSTAL,
NPCLCD= 11,
ALD=
    1.850,  2.333,  2.418,  2.907,  3.292,  3.544,  3.559,  3.962,
 3.966,  4.188,  4.209,
CL=
    0.3571,  0.4099,  0.4192,  0.4727,  0.5147,  0.5423,  0.5439,  0.5880,
 0.5884,  0.6128,  0.6150,
CD=
    0.0035,  0.0056,  0.0059,  0.0074,  0.0096,  0.0119,  0.0120,  0.0161,
 0.0162,  0.0184,  0.0186,
ALDA=
    3.701,  4.022,  4.119,  4.444,  4.757,  4.916,  4.952,  5.203,
 5.254,  5.370,  5.404,
CLA=
    0.3036,  0.3329,  0.3417,  0.3714,  0.3999,  0.4144,  0.4177,  0.4406,
 0.4452,  0.4558,  0.4588,
CDA=
    0.0331,  0.0361,  0.0370,  0.0401,  0.0430,  0.0445,  0.0449,  0.0472,
 0.0477,  0.0488,  0.0491,
```

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-185242		

4. Title and Subtitle	5. Report Date
High Speed Turboprop Aeroacoustic Study (Counterrotation) Volume II – Computer Programs	July 1990
	6. Performing Organization Code
7. Author(s) C.E. Whitfield, R. Mani, and P.R. Gliebe	8. Performing Organization Report No.
	10. Work Unit No. 535-05-01
9. Performing Organization Name and Address GE Aircraft Engines 1 Neumann Way P.O. Box 156301 Cincinnati, Ohio 45215-3191	11. Contract or Grant No. NAS3-23721
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191	13. Type of Report and Period Covered Contractor Report (Final)
	14. Sponsoring Agency Code

15. Supplementary Notes

Project Manager: Dr. James H. Dittmar, Propulsion Systems Division
NASA Lewis Research Center

16. Abstract

The isolated counterrotating high speed turboprop noise prediction program developed and funded by GE Aircraft Engines has been compared with model data taken in the GE Aircraft Engines Cell 41 anechoic facility, the Boeing Transonic Wind Tunnel, and in NASA Lewis Research Center's 8x6 and 9x15 wind tunnels. The predictions show good agreement with measured data under both low and high speed simulated flight conditions.

The installation effect model developed for single-rotation, high speed turboprops has been extended to include counterrotation. The additional effect of mounting a pylon upstream of the forward rotor has been included in the flow field modeling.

A nontraditional mechanism concerning the acoustic radiation from a propeller at angle of attach has been investigated. Predictions made using this approach show results that are in much closer agreement with measurement over a range of operating conditions than those obtained via traditional fluctuating force methods.

The isolated rotors and installation effects models have been combined into a single prediction program, results of which have been compared with data taken during the flight test of the B727/UDF®* engine demonstrator aircraft.

It is the satisfactory comparisons between prediction and measured data for the demonstrator airplane, together with the identification of a nontraditional radiation mechanism for propellers at angle of attack that constitute the major achievements of this Contract.

*UDF® is a registered trademark of the General Electric Company, U.S.A.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Counterrotation propeller noise, steady loading and thickness noise, installation effects on propeller noise, fuselage effects on propeller noise, aircraft control integration	or in part. Date for general release: August 1992.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified		